20 TIPS!

DOS WORLD

The Essential Guide to DOS and Windows

Sight & Sound

- ◆ **Multimedia** Without Windows
- ◆ Add **Sound** Cards, **Upgrade** Kits, & CD-ROMs
- ◆ Setting Up **Memory** For Multimedia

FREE PROGRAM!
Mosaics & Crazy Quilts in QBasic

COOL!
Great Music Without A Sound Card

HOW TO SPEED UP YOUR HARD DRIVE:
TIPS for SmartDrive and PCKwik

May 1995   No. 21 • $4.95
Canada $5.95

05 >

0 74470 78613 9

Safeguard Your System Settings

REMARKABLE! Faxing from DOS • Add 'Buttons' to DOS

Contents

NeoPaint Presentations, from NeoSoft Corp., page 80

How to Contact The Editors

DOS World *welcomes letters, complaints, and submissions from readers. The easiest way to reach the editors is the U.S. mail:* DOS World, *86 Elm St., Peterborough, NH 03458. All letters to the editor and questions are understood to be submitted for publication unless otherwise indicated. You can reach our staff electronically over CompuServe at 75300,2361. Please include your complete address and a daytime phone number on your correspondence.*

Also, you can reach the editors through the DW bulletin-board system (603-924-3181). To connect, set your modem and software to 8 data bits, no parity, 1 stop bit. DW's BBS lists all the QBasic and shareware programs mentioned in these pages. Shareware items listed on the DW BBS are products protected by copyright law. You're welcome to try these programs. If you find them useful, we ask you to register and pay the applicable fees to the programs' respective owners.

DW encourages the submission of DOS tips, QBasic programs, and batch-file articles from readers. Please submit all material on disk in ASCII format together with a double-spaced printed copy to our offices. The magazine cannot be responsible for the return of these manuscripts, unless an appropriate mailer and return postage are enclosed.

DW makes every effort to verify its articles and programs, but the magazine assumes no responsibility for any damages because of errors or omissions in our articles. Corrections to listings or articles are printed as soon as possible.

EDITORIAL

Multimania

by Michael J. Comendul

The word sounds like a Blue Light Special at Circuit City, but it's actually the only way to describe with any accuracy the misconceptions, often reflected in the computer press, about the emergence of multimedia on the PC. Why it has taken so long to come such a short way down this road is a mystery. Is it Microsoft's fault? After all, the company added its MSCDEX.EXE extension, an essential ingredient of CD-ROM set-up, only very recently, to the DOS 6.x series.

More likely, the regrettable state of multimedia is a result of the usual lack of cooperative standards among players: The world of PCs is more factional than an Italian parliament.

Is it a secret that multimedia on the PC is barely worth the price of admission? Peep-show video clips and a tune you can hum don't in themselves fulfill the promise of multimedia. But on your PC, they'll have to do.

Change, however, is on the way. New sound cards with wave-table synthesis are more common, and, more important, video graphics adapters with MPEG translators are becoming more mainstream.

MPEG is today's answer to television-like quality on a PC's screen, and we won't have anything that fulfills the promise of multimedia until the MPEG chip is commonly found on a PC video graphics adapter. Developed by the Motion Picture Experts Group, MPEG is a hardware-compression standard designed to capture video to CD-ROM (or any digital medium) and, conversely, decompress that signal for display on screen.

Once you've stored ample data for a screen image, the final hurdle is getting that data on screen. The PC can't process the 30 frames per second needed to produce television-like video. Today's video cards—even with accelerators—can't keep up with the massive data churn necessary to refresh the screen. That's why many of today's edutainment titles take shortcuts; they deliver only 10 to 15 frames per second of a video clip or animated sequence, giving images a jerky, robotic look. CD designers reduce the size of the screen you can see, which improves the speed at which you can see it. Further, they reduce the number of colors in an image's resolution, cutting the amount of data that needs to be conveyed to the screen, but corrupting the "continuous tone" of the image. That's why pictures on a PC often look *rasterized*: as though they're made up of noticeable digital blocks.

While video lags far behind in the multimedia sweepstakes, sound at least has taken enormous strides. Sound cards are leaving FM synthesis behind to make way for *wave tables*—actual sound samples of musical instruments—which have vastly improved PC music. And the inclusion of Microsoft's Media Player in Windows has at least put some practicality into CD-ROM, letting you use your drive to play audio compact discs.

Today, whether you choose to run DOS or Windows isn't the issue in multimedia computing. The issue is how to prepare for change. If you're investing in multimedia hardware and you don't want to be left behind, you'll want the fastest chips you can afford, a graphics accelerator card with MPEG on board running on a PCI bus, and an audio card with wave-table synthesis. ∎

Letters to the Editor

Help!

I have three questions I hope you can answer. First, do you have current addresses for Covox Inc. and Wenham Software? Though they're older products, Covox's Speech Thing and Wenham's Batcom are too good for me to simply let them wither on the vine.

Second, where does old, "obsolete" computer hardware go to die? Somewhere there must be warehouses full of equipment.

And, last but not least, when copyrighted software is no longer available from its copyright holder, what's the penalty for copying it to supply to others at no cost?

Homer Tilton
Tucson, Ariz.

We don't have current information on Covox Inc. or Wenham Software (readers?), but we can help out with your other questions.

*In January, the Boston Computer Society established a nationwide consortium for recycling computer equipment to nonprofit organizations. For further information, contact Marlene Archer via the Internet at **msa@bcs.org** or at BCS (101A First Ave., Waltham, MA 02154, 617-290-5700). If you're looking for a computer for your own use, try your local user group.*

Regarding copyrighted software, the Software Publishers Association (202-452-1600) says that it's not safe to assume that a company no longer promoting a product has no interest in protecting its copyright. The company may decide at a later date to sell the program to another firm, and companies that are going out of business frequently
sell their copyrights. If you want to check on the copyright status of a particular product, contact the U.S. Copyright Office (202-707-3000).

Incidentally, the SPA actively pursues parties it suspects are guilty of copyright infringement. Typically, fines amount to twice the value of the illegally copied software. —Eds.

A Little Cleanliness Goes a Long Way

The first hardware question in the September 1994 "Q&A" [*DOS World* #17, page 71] concerns a floppy drive that suddenly stopped reading disks. Technical Editor John Wolfskill's answer was to buy a new floppy drive.

I suggest that this should be the second step; I'd invest $5 in a head-cleaning kit first. As a volunteer who services computers for a nonprofit organization and recycles old PCs into inner-city schools, I've yet to find a "dead" floppy drive I couldn't revive with a simple head cleaning.

Harwood Loomis
Woodbridge, Conn.

Sound advice. To keep dust from accumulating, use a nonabrasive head-cleaning kit on your floppy drives at least once a month. Computer dealers and office-supply stores often carry such kits, which consist of a 3.5- or 5.25-inch disk and a bottle of cleaning fluid. —Eds.

Heading Headache

DOS World's format is good, and the content is important. Don't let your publication meet the same fate as *PC Magazine* and *PC Computing*, both of which started off
fine, but then turned into advertising journals that don't support the users of the equipment they showcase. By the way, what does *DOS World*'s section head "FYI" mean?

Bill Ripley
Memphis, Tenn.

Thanks for the praise, Bill, and, for your information, FYI *stands for just that.* —Eds.

Don't Let the Ozone Come Down on Me

In "Understanding Your PC's BIOS" [*DOS World* #17, September 1994, page 51], author John Bryan states: "Stray ultraviolet light, while it won't erase the [BIOS] chip, can change it enough to fail. While there's little random ultraviolet light bouncing around inside the average computer, of course, in these days of a deteriorating ozone layer it's best not to take chances."

Although I thought the rest of the article contained many good points, this portion was unacceptable to me. Ronald Bailey, in his book *Eco-Scam*, explains that in Antarctica, the ozone level fluctuates naturally: "Chlorine and bromine atoms and chlorine monoxides [are] free to attack [the] ozone when the returning sun peeks over the horizon at the beginning of the Antarctic spring in September and October.... Nearly all of the ozone between 12 and 25 kilometers is destroyed, allowing increased amounts of UV light to reach the surface. As summer approaches, the clouds disappear, the hole is filled with newly produced ozone . . . and UV levels return to normal."

As for computerists in North America, we have nothing to worry about, ozone-wise.

Alex Bischoff
Pittsburgh, Pa.

4DOS Fan

I've been reading your magazine since the first issue. I've noticed readers' letters and contributors' articles about creating batch files to perform various tasks and using ANSI escape sequences to control the screen, cursor, and so on. In all this time, I've been disappointed that you haven't advocated the use of JP Software's 4DOS program as an easier solution.

4DOS (version 5.5c is the latest) is available on many BBSes as shareware, and in my opinion it's a superior replacement for MS-DOS's COMMAND.COM and a terrific substitute for the rest of MS-DOS's commands and external programs.

In fact, quite a few people have questioned Microsoft's wisdom in not licensing 4DOS for incorporation into MS-DOS itself. I think you should devote a major portion of a future *DOS World* issue to 4DOS.

Jerome T. Czeikus
Victorville, Calif.

For those readers who missed it, we published an article devoted to 4DOS, "A Better DOS Than DOS," in the January 1995 issue (#19, page 48). —Eds.

Fun Facts to Know and Tell

Dan Gookin's "Test Your DOS IQ" [*DOS World* #18, November 1994, page 53] was most interesting. Under "DOS Lore and Legend," however, let's not forget TRS-DOS, used by Radio Shack's Tandy TRS-80 models, and DOSPLUS, developed by Digital Research for the GEM GUI. And then there's CP/M. It, too, was developed by Digital Research and is the basis for the DOS we know today.

Dibrell L. DuVal
Tulsa, Okla.

ERRORS AND OMISSIONS

Fixer-Uppers

Sharp-eyed readers have uncovered several mistakes in the January 1995 issue of *DOS World* (#19):

● In "Disaster Averted" (page 22), we give incorrect information about how to make the DEL command always prompt you before deleting a file. You can't use DIRCMD to force the /P switch for DEL. You can do it, though, by adding the following DOSKEY command to your AUTOEXEC.BAT file:

```
DOSKEY DEL=DEL $1 /P
```

● In "A Batch-File Breakthrough" (page 36), following the author's lead and using a variable named TEMP in EXTRACT.BAT may cause problems for some users. Many people use this variable for other purposes, adding a line such as this one to their AUTOEXEC.BAT files:

```
TEMP=C:\TEMP
```

This line tells DOS, Windows, and many applications where you want to store temporary files. If your AUTOEXEC.BAT contains a TEMP= command or you use TEMP in some other way, running EXTRACT.BAT wipes out this setting, which could make DOS behave strangely until you reboot. To avoid trouble, substitute another variable name.

● The tip "Take the Short Road" (page 50) states incorrectly that the command DIR /O:G produces an alphabetical list of files, with directories first. It does list directories first, but neither directories nor files are in alphabetical order. The correct command for doing this is DIR /O. (You may also use DIR /O:GN; the GN is superfluous, but doesn't trigger an error message.)

● In Q&A (page 66), the discussion regarding databases and the "Access denied" message contains an incorrect ATTRIB command. The final command in the second column should read as follows:

```
ATTRIB filename -R -H -S
```
—Eds.

Midnight Blues

John Wolfskill's batch file WINDAY.BAT, which backs up files after ending a Windows session ["Q&A," *DOS World* #19, January 1995, page 67], contains a subtle bug.

As written, the program will cause problems if you start Windows before midnight, create a few new files, then quit Windows after midnight. When it executes the XCOPY command, WINDAY.BAT uses the current system date, which was updated at midnight. Because the Windows files saved before midnight have yesterday's date, they aren't copied.

To fix this problem, you must rearrange the batch file to save the date—in a file or in an environment variable—before running Windows. After exiting Windows, use the saved date in the XCOPY command.

Richard Penn
Montreal, Quebec
Canada

Good catch, and there's a simple fix. Move these two commands:

```
CD C:\WINDOWS
WIN
```

from the top of the program, placing them after this line in WINDAY.BAT:

```
DEL GETDATE.BAT
```
—Eds.

Bulletin Board Is the Best

Consider this a note of thanks for the excellent service you provide on the *DOS World* bulletin-board system [603-924-3181]. Your magazine has been a lifesaver, and the BBS complements it. I'd hate to be without the BBS; I recommend it to everyone I know, even those Windows wretches.

Marshall Hubbard
Memphis, Tenn.

Tips from Readers

AUTOEXEC.BAT Improvements

Your readers may be interested in three simple routines I include in my AUTOEXEC.BAT file.

First, I use the following DOSKEY command to set up a command called DD:

```
C:\DOS\DOSKEY DD=DIR $* /O-D /P
```

This line uses the DIR command to list files in the current directory, displaying the most recent files first. I find DD helpful for figuring out the names of the files on which I've worked recently.

Including the sequence $* lets the command process any number of parameters, which in turn allows me to issue commands such as the following:

```
DD *.BAS
```

The /P switch in the DOSKEY command (at top) tells DOS to pause after displaying each page of the directory. I can then press any key to continue, or press the Break key to return to the DOS prompt.

My second addition to AUTOEXEC-.BAT is also a DOSKEY command:

```
C:\DOS\DOSKEY CDD=CD..\$1
```

This command helps me navigate the subdirectories below my QuickBasic directory.

CDD comes in particularly handy, for instance, when I want to keep two versions of a program, one of which will reside in QB\STOCK and another in QB\GAMES.

To jump between the two directories in one step, I simply type CDD GAMES or CDD STOCK.

My third trick is a PROMPT command I use with QuickBasic and other programs that let me temporarily exit to DOS, do some work, and then return to the original application by typing EXIT:

```
PROMPT $E[Ø;133;"EXIT";13p$P$G
```

With this PROMPT command in place, instead of entering EXIT, I press the F11 key, which I've reprogrammed to type EXIT and execute a carriage return.

George Dunbar
Calgary, Alberta
Canada

Belling the Batch

In describing his disk-formatting batch file FMAT.BAT ["Q&A," *DOS World* #18, November 1994, page 66], John Wolfskill suggests creating the program at the DOS prompt, because most word processors and text editors don't let you enter the BEL character, ASCII 7, from the keyboard.

Readers who have occasion to use a BEL character in several of their batch files might instead want to create a batch file called BELL.BAT to echo the bell.

When you use the COPY CON method outlined by Wolfskill, the batch file looks like this:

```
@ECHO OFF
ECHO ^G
```

To produce ^G, press and hold the Alt key while typing ØØ7 on the numeric keypad.

If you prefer to work in EDIT.COM instead, your batch file will look like this:

```
@ECHO OFF
ECHO ●
```

You can produce the bullet, the ASCII equivalent of the BEL character, by pressing Ctrl+P, then Alt+ØØ7, or by pressing Ctrl+P, then Ctrl+G. Both versions of the batch file do the same thing.

Whenever you want to include a warning tone in another batch file, simply add the command CALL BELL to your program. (Note that BELL-.BAT must be in the current directory or a directory mentioned in your PATH statement.)

Michael Bright
Tampa, Fla.

A Really Watchful Eye

There's a more efficient way to send the date or time to a file than the one described in the tip "A Watchful Eye" [*DOS World* #19, January 1995, page 64]. Substitute these two lines for the second and third lines of ACCESS.BAT:

```
ECHO. |DATE|FIND "C" >> ACCESS.LOG
ECHO. |TIME|FIND "C" >> ACCESS.LOG
```

By using ECHO. to send a carriage return/linefeed, you eliminate the need for CRLF.DAT and save yourself disk space, which may amount to as much as 8192 bytes.

In addition, you may have FIND look only for *C*. It will echo the entire line containing that phrase.

Tom Dye
Vineland, N.J.

One other modification is worth noting. The text suggests that to read your access log you add the command ACCESS to your AUTO-

EXEC.BAT file. This works as long as the command is the last line in your AUTOEXEC.BAT. *If you place the command elsewhere, though, use* CALL ACCESS *instead. Otherwise, DOS won't execute any of the commands in* AUTOEXEC.BAT *that follow the* ACCESS *command.* —Eds.

Conserving Memory

I used to start the following DOSKEY file-finder macro from my AUTOEXEC.BAT file (L4 stands for *look for*):

```
LOADHIGH DOSKEY L4=DIR $1*.* /S /P
```

But because loading the macro high tied up 4K of memory, I then decided to use a batch file, L4.BAT, instead:

```
DIR %1*.* /S /P
```

L4.BAT doesn't use up any memory, except when it's running. To ensure that DOS can locate my file finder, I mention the batch file's directory in my PATH statement.

An interesting feature of L4 is that it displays a directory of an entire drive when you type L4 *.*.
Roy Reed
Tampa, Fla.

Escape Esc

If you're like me, you hate having to press the Esc key to clear the

screen every time you start QBasic. Here's how to avoid that little annoyance.

If you want to start QBasic and run a program, use the following syntax:

```
QBASIC /RUN program
```

To start QBasic and load a program you want to edit, use this command instead:

QBASIC *program*

You can use the latter command even if the program doesn't exist yet. QBasic will start, clear the immediate window, and display the program name you provided, showing it in the box at the top center of the window. Later, you may use the Save As option to change the program's name.

You can use an analogous command with DOS's Editor program, EDIT.COM, which is actually part of QBasic. (For more on using EDIT.COM, watch for our feature "Unearth the Editor" in an upcoming issue of *DOS World*.)

For instance, to start right in creating a file called NEWNEW, just type this line:

```
EDIT NEWNEW
```

Note that you may include a filename extension if you like.
Lane Olinghouse
Everett, Wash

Undo All Over Again

When you're editing a document in a program that offers an Undo option, remember that you can toggle between the original version and the replacement by repeatedly choosing Undo.

If you're working in Windows Write, for example, highlight the word, phrase, or paragraph you want to change or replace and type in your alterations.

Then choose the Undo command to see the original version of the document. When you choose Undo again, the changes you made will return. (Note that not all applications offer this feature.)

This sequence lets you compare the two options as many times as necessary and decide which one you prefer before advancing to the next round of editing.
Lane Olinghouse
Everett, Wash.

STRANGE BUT TRUE

Because the SUBST command lets you substitute one drive letter for another drive and path, it's useful for shortening the PATH statement in your AUTO-EXEC.BAT file. But if you make a lot of substitutions, you can easily forget which drive letters are assigned to which drives. To find out without having to scan AUTOEXEC.BAT, issue this undocumented command at the DOS prompt:

```
TRUENAME drive:
```

where *drive* is the letter of the drive for which you want DOS to reveal the true directory name.

—Jack Nimersheim

READER FORUM

The Answer ...
And the Real Answer

by Hardin Brothers

I had a teacher once who said that most questions could be answered in three ways: You could tell someone the answer directly, you could explain the reasons behind the answer, or you could tell someone how to find the answer.

For example, if someone asked how to bake a loaf of bread, the straightforward answer would be a recipe. The explanatory answer would include information about each step of the process—why the yeast requires sugar, why the dough has to be kneaded, why it has to sit and rise before baking. The third answer would point to books about bread, or the bread sections of standard cookbooks.

Most of the questions I receive for this column can be answered in a similar fashion. Readers who are frustrated by computing problems will probably appreciate straightforward answers most. Those who want to better understand their computers appreciate explanatory answers. And those with similar problems may benefit most from reference material. I'll try to give all three types of answers to each question whenever possible.

Inserting ESC

Jack Nelson of New Orleans, Louisiana, starts us off with a question that has plagued many users:

I enjoyed Doug Lowe's "Pros' Choice: A Wealth of Options" in DOS World *#17 [September 1994, page 39]. The article described how to use the CHOICE command to write a menu batch file and how to use the /C:KEYS switch to limit the user's response. One of the* menus used the escape marker to return to DOS. My problem is that I've been unable to create this batch file because I can't embed the escape marker in a file. I've tried using DOS's EDIT.COM, along with a couple of other editors. Please tell me what key or key combination you use to create the left-pointing arrow representing the escape marker.

> *I'll try to give you as much information as possible: yes or no, how and why, and where to go to learn more.*
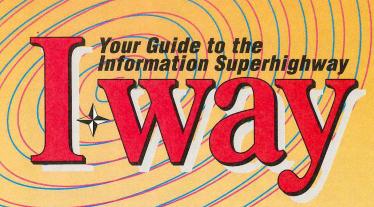
The short answer is simple: In EDIT.COM, press Ctrl+P and then the Esc key. When you press Ctrl+P, EDIT.COM will display ^P on its status bar. DOS will insert into your document the next key you press, instead of interpreting that key as an EDIT.COM command. Most editors include a similar (often identical) mechanism for inserting special key codes. EDIT.COM's use of ^P, along with many of its key commands, is adapted from WordStar, one of the first popular word processors for DOS-based computers.

It should be easy to find this information in EDIT.COM's on-line help—but it isn't. The information is there, but you have to look for it. First, select Help/Keyboard. Now you'd guess that you want Insert and Copy Keys. Look at the help screen carefully. On the last line, you'll see "Insert special characters Ctrl+P, Ctrl+*key*."

You may recognize that as the help you need—at least until you try it. Ctrl+P, Ctrl+A does seem to work; it puts a little happy face on screen. But Ctrl+P, Ctrl+Esc doesn't seem to do anything at all. To understand why that doesn't work, you need to know a little about key codes.

EDIT.COM lets you enter any of the 95 printable ASCII characters (32 through 126). In computing,

ASCII—which stands for *American Standard Code for Information Interchange*—represents characters as numbers. When using EDIT.COM, for example, you get a character by holding down the Alt key and typing the corresponding digits on your numeric keypad (Num Lock on): 32 for a space, for example, or 126 for a tilde (~). When you release the Alt key, the character appears. EDIT.COM also lets you enter "extended" ASCII characters: ASCII 128 to 255.

But it's harder to enter codes 0 (zero) through 31, which represent keys that EDIT.COM usually interprets as commands: the Enter key, the Esc key, all the Ctrl+*letter* keys, and so on. If you want a smiley face (ASCII code 1) to appear in your file, you know you have to enter Ctrl+A, but if you do just that, EDIT.COM interprets it as the command to move the cursor one word to the left.

The Ctrl+P key combination is a special EDIT.COM command. It says, "Insert the following key into the document—don't interpret it as a command." So the correct combination for the escape marker is Ctrl+P, Esc. If you want to try something a little more esoteric, try Ctrl+P, Ctrl+[or Ctrl+P, Alt+27.

By the way, the Ctrl+P command isn't without bugs, at least in the version of EDIT.COM on my machine. I can't use it to insert Ctrl+J, Ctrl+L, Ctrl+M, or Ctrl+U (ASCII codes 10, 12, 13, and 21) into a document. Anyone want to guess why they won't work?

Leaving QBasic

Chris Turnbull of Waterloo, Ontario, Canada, has a common problem with QBasic:

In "Readers' Queue," DOS World #17 [September 1994, page 66], Tom Sales presented three one-line QBasic programs. I entered them into my computer and enjoy them thoroughly. But how do you exit from them without doing a warm boot? I don't use QBasic very much except for the occasional program I type in. These three programs are the first I can't get out of.

You can almost always stop a QBasic program by pressing Ctrl+Break. Then you can return to DOS by selecting Exit from the File menu. As far as I know, Ctrl+Break isn't documented in QBasic's on-line help. It's just something that has been included in various versions of Basic for several years; experienced users expect it to work, and it does. If you don't have the benefit of experience, you'll find this feature only by trial and error, or by asking the right question at the right time.

QBasic includes other undocumented features, as well. For example, according to the on-line help, the SYSTEM command closes all open files and returns control to the operating system. That description is

correct—sometimes. SYSTEM works that way if you start a QBasic program with the command QBASIC /RUN *program*. But if you start QBasic, load your program into its editor, and then get the program running, SYSTEM will close all open files and return you to the QBasic editing screen.

If you want to learn about QBasic's undocumented and underdocumented features, you'll need to ask other QBasic users. I don't know of any single repository for QBasic lore and techniques.

The Wrong Drive

The final question this month comes from Bruce Sharp of Des Moines, Iowa:

I tried to follow your instructions in "In Search of a Missing Menu" ["Letters to the Editor," DOS World #18, November 1994, page 7] to install the Tools menu for Windows' File Manager to view the DoubleSpace Info option. I also followed instructions in DOS 6.2's README.NOW file for adding options after upgrading to that version of DOS. I have a 5.25-inch drive as drive A on my computer, but my DOS disks are on 3.5-inch floppies. I made a boot disk, restarted the computer, typed PATH C:\DOS and then changed to drive B, which held the first DOS installation disk. I typed SETUP /E and expected to see the installation menu, but the only thing I saw was this message: "To install MS-DOS, insert Disk 1 in drive A: and press Ctrl+Alt+Del." Can you help?

Your version of DOS is different from the one described in the article. The author was using a DOS upgrade, which is available at computer stores. You have a new DOS installation kit, which is usually available only from manufacturers.

You can install your version of DOS on a blank hard drive. Therefore, it needs to load the core of its own operating system so that it can run its setup program. And the only way it knows how to do that is to insist that you boot from the first installation disk.

Want to Know More?

If you have a question, or need further help with something that has appeared in DW, *write:*

Reader Forum
DOS World
86 Elm St.
Peterborough, NH 03458

Please include a copy of your AUTOEXEC.BAT and CONFIG.SYS files on disk if you have particular problems with software or hardware conflicts and compatibility.

If you had DOS on 5.25-inch disks, or if you could boot from your 3.5-inch drive, you'd be all set. Buying a DOS upgrade on 5.25-inch disks costs a few dollars—and is unnecessary if you're willing to do a little work.

I know of at least two ways to boot from your 3.5-inch drive. The first and easiest is to run your computer's BIOS setup program. During boot-up, you probably see a message that says something like "Press Alt+F1 to run SETUP." On some 80286 machines, you have to run a special setup program from disk.

If you haven't done so already, print the main information screen or write down the information it displays about your hard drives. You'll need it if your computer's battery fails.

Now look for an option to select the boot floppy. Some setup programs display this option on the first page. Others place it under an "Advanced Setup" or "Other Options" heading. And on some computers, you won't find it at all.

If you do find such an option, tell Setup that you want your second disk drive to be your boot floppy. Then restart your computer, and watch the lights to see whether the 3.5-inch drive looks for a boot disk. If it does, you're all set.

If you can't change drives from the BIOS Setup menus, you're still not sunk. Turn off your computer, unplug it, open the case, and find the cable that connects both drives to the disk controller board or the motherboard. It's a broad, flat cable—probably gray. Use a little paint or fingernail polish to mark which connector goes to which drive and which side of each connector is up. Remove the cable and plug in the 3.5-inch drive so that it's first on the cable; then plug in the 5.25-inch drive so that it's second.

Close the computer, plug it in, and turn it on. Run Setup to properly identify drives A and B. When you watch the drive lights, you should see the one on the 3.5-inch drive come on first. Remember that it's drive A and that the 5.25-inch drive is B. Now you can boot from a 3.5-inch drive.

You can leave the drives configured "backwards" if you want. (I did that on my computer after I switched them to install OS/2.) But you and everyone else who uses your computer will have to adjust to the changed drive letters.

Once your computer boots from your 3.5-inch drive, make sure you save a copy of your AUTOEXEC.BAT and CONFIG.SYS files. Then run your DOS setup program and follow the option menus. For more on system setup, see "Without You I'm Nothing," page 28 in this issue, as well as this month's "Start-Up Clinic," page 63. ■

Technical Editor Hardin Brothers has been working with computers and writing about them for 15 years.

Best of the Batch

PATH Ways

PE.BAT (see the accompanying listing, right) is a batch file that lets you easily add and delete subdirectories in your path. What makes this program particularly useful is its variety of options. It can do any of the following:

- add a subdirectory to the end of your current path
- add a subdirectory by inserting it anywhere in your path
- add a subdirectory by substituting it for another subdirectory
- delete any subdirectory in the path
- move a subdirectory from one position in your path to another

The accompanying table (facing page, bottom) shows you the syntax for each operation.

One of PE.BAT's nicer features is its display. Rather than list a path as one long string, as AUTOEXEC.BAT presents it, PE.BAT gives each subdirectory its own line. For example, let's say your original subdirectory looks something like this:

```
C:\DOS;C:\WINDOWS;C:\WINWORD;C:\TEMP;C:\UTILS
```

If you decide to insert the subdirectory C:\BATCH before C:\DOS with the following command:

```
PE + C:\BATCH C:\DOS
```

PE.BAT will give you the following display:

```
** Path is currently set to . . .
C:\DOS
C:\WINDOWS
C:\WINWORD
C:\TEMP
C:\UTILS

** Adding (C:\BATCH) at (C:\DOS) . . .

** Path is now SET to . . .
C:\BATCH
C:\DOS
C:\WINDOWS
C:\WINWORD
C:\TEMP
C:\UTILS
```

The batch file PE.BAT not only lets you add and delete subdirectories from your current path, it lets you insert and replace them, as well.

```
@ECHO OFF
IF %proname%. == . SET proname=%0
IF %1. == ``. GOTO display
IF NOT %pe%. == . GOTO start
ECHO:
ECHO ** Path is currently set to ...
CALL %proname% `` %path%
ECHO.
:start
IF %1. == . GOTO end2
IF %1. == `+. GOTO add2
IF %1. == `-. GOTO del2
IF %2. == . GOTO info
IF %1. == +. GOTO add
IF %1. == -. GOTO del
GOTO info
:add
IF %3. == . ECHO ** Adding (%2)...
IF not %3. == . ECHO ** Adding (%2) at (%3)...
ECHO:
SET pe=%2
SET mv=%3
%proname% `+ %path%
:add2
SET path2=%path%
SET path=
:add-loop
SHIFT
IF %1. == . GOTO add-out
IF %1. == %mv%. SET add=added
IF %1. == %mv%. SET path=%path%;%pe%
IF %1. == %pe%. GOTO add-loop
SET path=%path%;%1
GOTO add-loop
:add-out
IF %add%. == added. GOTO end
IF not %mv%. == . GOTO add-not
SET path=%path%;%pe%
GOTO end
```

Continued opposite

Note that the subdirectories you type as parameters must be in uppercase. If you need help, type PE ? and the program will show you the proper syntax for each operation.

Bobby Gage II
Frederick, Md.

```
:add-not
ECHO ** (%mv%) Does not exist...
ECHO:
SET path=%path2%
GOTO end2
:del
IF %3. == . ECHO ** Deleting (%2)...
IF not %3. == . ECHO ** Replacing (%2) with (%3)...
ECHO:
SET pe=%2
SET mv=%3
IF %pe%. == %mv%. GOTO d-error
%proname% `- %path%
:del2
SET path=
:d-loop
SHIFT
IF %1. == %mv%. GOTO d-loop
IF NOT %1. == %pe%. SET path=%path%;%1
IF %1. == %pe%. SET pe-d=deleted
IF %1. == %pe%. SET path=%path%;%mv%
IF %2. == . GOTO end_delet
GOTO d-loop
:end_delet
IF NOT %pe-d%. == . GOTO end
ECHO ** (%PE%) Does not exist !!
ECHO:
GOTO end2
:d-error
ECHO ** (%pe%) is the same as (%mv%)
ECHO:
GOTO end2
:info
CLS
ECHO.
ECHO.
ECHO SYNTAX
ECHO:
ECHO (1)  pe + path      - add to the current path
  .......%proname% + C:\WP51
ECHO (1a) pe + path path - add or move path to
  ...........%proname% + C:\WP51 C:\DOS
ECHO (2)  pe - path      - delete the path
  ..............%proname% - C:\WP51
ECHO (2a) pe - path path - delete the path and
  replace...%proname% - C:\WP51 C:\WP
```

```
ECHO (3)  pe ?           - display this help
  .............%proname% ?
ECHO (4)  pe             - display the current path
  .....%proname%
ECHO:
ECHO (1)  Will add C:\WP51 to the current path.
ECHO (1a) Will add C:\WP51 before C:\DOS.
ECHO (2)  Will delete C:\WP51 from the current path.
ECHO (2a) Will delete C:\WP51 and replace it with
  C:\WP.
ECHO (3)  Will display help.
ECHO (4)  Will display the current path setting.
ECHO:
ECHO Note: (C:\WP51,C:\DOS,C:\WP) are example
  directories.
ECHO:
GOTO end2
:display
SHIFT
IF %1.== . GOTO end3
ECHO %1
GOTO display
:end
ECHO ** Path is now SET to....
CALL %proname% `` %path%
ECHO:
:end2
SET add=
SET mv=
SET pe=
SET pe-d=
SET proname=
SET path2=
:end3
```

End

PE.BAT SYNTAX

Operation	Original Path	PE Command	New Path
add subdirectory	C:\DOS;C:\WINDOWS	PE + C:\TEMP	C:\DOS;C:\WINDOWS;C:\TEMP
insert subdirectory	C:\DOS;C:\WINDOWS	PE + C:\TEMP C:\DOS	C:\TEMP;C:\DOS;C:\WINDOWS
delete subdirectory	C:\DOS;C:\WINDOWS	PE - C:\DOS	C:\WINDOWS
replace subdirectory	C:\DOS;C:\WINDOWS	PE - C:\DOS C:\TEMP	C:\TEMP;C:\WINDOWS
move subdirectory	C:\DOS;C:\WINDOWS	PE + C:\WINDOWS C:\DOS	C:\WINDOWS;C:\DOS
current path		PE	
help		PE ?	

SHAREWARE EXCHANGE

Quiet Performers With a Big Finish

by Hardin Brothers

Who needs bells and whistles when simpler can be better? That's clearly the attitude among the *DOS World* readers who contributed to this month's shareware nominations. At work or at play, you can get the job done right with no-frills shareware programs available on the *DOS World* bulletin-board system (603-924-3181), as well as from most information services and local BBSes. (See the sidebar, "Share the Wealth," page 22, and the *"DOS World* BBS" section of "How to Use This Magazine," page 72, for details on using the bulletin board. Prices are listed in the "Product Information" box, page 22.)

> *For work or play, simpler can be better. Get the job done right— thanks to economical, no-frills shareware programs.*

No-Fuss Telecomm
Carl Keehn of Eau Claire, Wisconsin, was looking for a small, easy communications program. That's exactly what he found with **Lync**:

I downloaded Lync 3.0 from a bulletin board and found that it's a great communications program. You can get in and out faster than you can with its rival shareware programs, and it takes up only 56K of space on your hard drive after you print and then delete the owner's manual.

I enjoyed Lync 3.0. (Look for LYNC30.ZIP on the *DOS World* BBS.) It's fast and easy to use, yet contains the essential features you need for hooking up with bulletin boards. You can also set it up in host mode as a mini-BBS, but you probably won't want to, because it doesn't offer security features.

Lync supports X-modem, Y-modem, and Z-modem protocols, which will handle most file transfers to and from BBSes. In general, Lync is a great communications program for a laptop computer with limited hard-drive space. For desktop use, however, I'd look for a program with a few more features, even if it consumes a little more disk space.

I was disappointed that Lync doesn't support 43- or 50-line screen mode, which I always use on my desktop computer. Also, it doesn't include any options to set terminal emulation type. Lync was last revised in 1992; since then many things have changed in the communications world. For example, Lync supports 9600-bps (bits per second) transmission, but not the popular 14,400-bps speed of V.32bis modems. Nevertheless, Lync is a good comm program worth its registration price, especially if you want to conserve hard-disk space.

A Shell of a Different Color
Benjamin J. Martnick of Wind Gap, Pennsylvania, isn't looking for spice. It's features he's after, and he's found what he needs in a text-based shell called **Elftree** (ET301.ZIP on the *DOS World* BBS):

I use Elftree 3.01 more than any other DOS, Windows, or OS/2 program. It's a file, directory, and program manager. You can do everything you need without leaving it: move, copy, and delete files; view files of all types, including graphics and sound files; and more. If DOS doesn't include a command to carry out a certain operation, you can create menus to call the

appropriate programs or DOS commands from within Elftree. The program works great with CD-ROM discs. It lets you attach notes to files, and you can customize it to your heart's content.

I should start by saying that I'm not particularly fond of shells and file managers. Several years ago, I found one I could tolerate, and since then I haven't found anything that comes close.

But even if I don't plan to use Elftree, I can appreciate what it does. The program offers everything from a file organizer to a screen blanker. You can use it to manage files, directories, and programs. In fact, it can do almost anything except multitasking.

Elftree's plethora of features comes at a price, however. Expect to spend time reading the manual and learning to use Elftree. If you simply try to understand the program from its on-screen menus, you'll miss many of its features. Because some of the program's capabilities change according to mode, it takes time to understand all of Elftree's ins and outs.

As I tested Elftree, I uncovered one problem. When I asked the program for a directory tree of a drive created with SUBST (drive F, in this case), it created a tree of the underlying host drive, D. Because it kept trying to add F: in front of names that referred to drive D, it got lost looking for files and directories. I never completely overcame this problem, but I did find a way to work around it satisfactorily.

Elftree doesn't excite me, but I don't think any program of this type would. I do appreciate its wide range of functions and its clean-looking displays, though. If you're in the market for an advanced shell, and you're willing to spend time learning to use it, Elftree is an excellent choice.

Low-Cost Text Editing

Donald Schmitz of Milwaukee, Wisconsin, has nominated **The Aurora Editor** (AURORA2.ZIP on the *DOS World* BBS), an excellent shareware choice for working with text:

The Aurora Editor is highly configurable and offers enough features to satisfy the most discriminating user. It provides many more tools than DOS's EDIT.COM *utility, including the ability to handle large text files. I loaded a 13,000-line listing, and Aurora took every bit of it.*

I've had fun working with The Aurora Editor, although I grew a little impatient while its 200 pages of documentation churned through my printer.

This editor has almost everything. It can use up to a gigabyte (1024MB) of virtual-memory space—a combination of XMS (extended), EMS (expanded), and disk space—so it can edit multiple files of almost unlimited size. It's fast, it features multiple windows, and it offers complete mouse support. It includes unlimited undo and redo.

Aurora provides bookmarks so that you can jump quickly to predetermined locations in your files, and it lets you redefine the pull-down menus. It can also substitute text automatically as you type. For example, if you define on-the-fly abbreviations, you can type ABQ and have Aurora translate it automatically into the word *Albuquerque* on screen. Or you can list your most common spelling errors (such as *teh*) and have it change each one to the correct word (*the*) automatically.

One unusual feature of Aurora is that it can "fold" a document to hide lines in which you're not currently interested. You can then view other parts of the document without the distraction of scrolling back and forth. Like most of Aurora, the fold function is fast, easy to use, and powerful.

You can use Aurora either for word processing or for working with text files (such as AUTOEXEC.BAT and CONFIG.SYS) or program source code. You can also use it as a binary editor, which means you can make direct changes to files that aren't in ASCII format.

What's missing? Not much. Aurora doesn't handle the fancy fonts and embedded graphics of high-end word processors. And it doesn't include a built-in spelling checker or thesaurus. But its file manager is almost good enough to use as a DOS shell, and it may be one of the most configurable text editors around. You can assign any internal function to any keystroke—an option available in several expensive text editors. You can also use its powerful built-in macro language to create keyboard shortcuts that will do almost anything you can think of.

If mention of Aurora's object-oriented, LISP-like macro language intimidates you, don't worry. You'll get your money's worth even if you never try to write a complex macro function. And you'll still be able to use its macros to record and play back keystrokes.

I'm impressed. Aurora offers as many functions and is as easy to use as the professional programmer's editor for which I paid $200. Unless you think DOS's EDIT.COM provides all the power you'll ever need, you should at least take Aurora for a test drive.

Running Circles Around Reversi

A friend who knows my predilection for games that require strategy and puzzle-solving skills gave me

SHARE THE WEALTH

Shareware Exchange" invites you to send a copy of your favorite shareware program, along with a description and an explanation of why you like it (500 words or less) to Shareware Editor, *DOS World*, 86 Elm Street, Peterborough, NH 03458. Tell us how we can obtain a copy of the program. We'll pay contributors $50 for each program featured in this column. Please don't send us shareware you've written yourself—we want recommendations from users, not authors.

You can contact us on CompuServe at 75300,2361 or on the *DOS World* BBS at 603-924-3181. All programs featured in "Shareware Exchange" are available electronically from the *DOS World* BBS (instructions on page 72). Lync (LYNC30.ZIP) is in File Area 3; NumLo (NUMLO10.ZIP), Ant Run (ANTRUN20.ZIP), and Loader Larry (LLARRY11.ZIP) are in File Area 4; and ElfTree (ET301.ZIP) and The Aurora Editor (AURORA2.ZIP) are in File Area 5. In addition, most of the programs described in "Shareware Exchange" can be found on major on-line services and local BBSes.

—Eds.

this last program. I don't usually write about games here, but I think many readers will enjoy **NumLo** (NUMLO10.ZIP on the *DOS World* BBS). It's an excellent addition to any game library.

NumLo is similar to the classic board game Reversi, which is marketed in the U.S. as Othello. The object of Reversi is to finish the game with more of your color circles on the board than your opponent has. You try to capture your opponent's circles by "surrounding" them.

But NumLo has several important differences. Imagine Reversi played on a 12-by-12-square board (rather an 8-by-8) with three players instead of two. Added to that are numbered squares that count for more than one point. Unlike in Reversi, not all squares are equal, so your strategy for protecting your squares and surrounding your opponent's circles must also include the value of each square.

In NumLo, you take the role of one contestant, and the computer takes the part of the other two players. Unlike many three-handed games, which degenerate into lopsided contests of two players versus one, NumLo always retains the flavor of three players working independently.

I enjoyed NumLo so much that I looked for other games by the same author, and I found more than 30 in CompuServe's IBMNEW forum. You can download three from *DOS World*'s BBS: NumLo, **Loader**

Larry (LLARRY11.ZIP), and **Ant Run** (ANTRUN20.ZIP). Loader Larry presents several mazes you solve by moving objects around. In Ant Run, you create and continually modify a series of passageways before that little critter gets to them. The object is to create as long a path as possible.

The presentation of all three games is more professional than that of many other shareware games, and play is better than that of some commercial games I've tried. They're short enough that you won't feel guilty playing a game when you want to take a break from more serious work. If you enjoy the mental exercise of a good strategy game, try all three! ∎

Technical Editor Hardin Brothers has been working with computers and writing about them for 15 years.

PRODUCT INFORMATION

Prices shown are for registration only. Most vendors charge additional small fees for shipping and handling, extra disk copies, and printed documentation.

Ant Run 2.0, $12
Loader Larry 1.1, $12
NumLo 1.0, $12
Soleau Software
163 Amsterdam Ave.
Suite 213
New York, NY 10023
212-721-2361

The Aurora Editor 2.0, $44.95
Aurora Terra
P.O. Box 34275
Bethesda, MD 20827
301-468-2255

Elftree 3.01s, $50
Alan J. Avery
Elvish Consulting
1408 Noble Ave.
Springfield, IL 62704
217-698-8600
includes file viewers and editor

Lync 3.0
$25 3.5-inch, $26 5.25-inch
Jason R. Alward
1607 Claymor Ave.
Ottawa, Ontario
Canada K2C1T4
mail inquiries only

HARD-DISK ORGANIZATION

Managing the Maze

Does your directory structure resemble the family tree from a daytime soap opera? If so, it's time to do some constructive pruning. Here are a few techniques to help you get your program and data files under control.

by Kenneth E. Johnson

The average computer today has a 400MB hard drive, which might seem like all the space in the world compared with the hard drives of a few years ago. But if you're not careful, that space can easily get cluttered with a confusing array of files and directories, making the hard drive difficult to navigate and files a challenge to locate. And some of that space can be wasted on duplicate files, old backups, outdated programs, and "temporary" files that have somehow become permanent.

If this sounds like an accurate description of your computer, you need to bring some management to your directory maze. And even if you're not living through a hard-drive nightmare now, a little planning on your part will guarantee that you'll never have to.

Contributing Editor Ken Johnson is training and support manager at the law firm of Mayer, Brown & Platt in Chicago. He's a columnist and writer for several business and computer-law publications and serves as a contributing editor for the American Bar Association's NETWORK 2d newsletter.

The more you compartmentalize files, the easier it is to find them when you need them. The more file duplicates and old programs you delete, the more disk space you open up for your data and new applications. Let's look at some organizational techniques that will help get those wanton directories under control.

Keep the Root Directory Clean

Start your disk organization by keeping your root directory small. One sure sign of a disorganized hard drive is a root directory jammed with files. The root directory should contain only a few necessary system files, including the hidden DOS files IO.SYS and MS-DOS.SYS (or IBMBIO.COM and IBM-DOS.COM for PC DOS), MS-DOS 6.x's DBLSPACE.BIN (also hidden), and AUTOEXEC.BAT and CONFIG.SYS.

Most people include COMMAND-.COM in the root directory also, but it's not absolutely necessary. The SHELL= statement in CONFIG.SYS can tell DOS where to find COMMAND-.COM if it's not in the root directory.

There's a practical reason for not filling the root directory with pro-gram and data files. DOS limits the number of files that can exist in the root directory of any disk, based on disk size. On most hard drives the limit is 512 files in the root. DOS places no limit, however, on the number of files a subdirectory can hold. Subdirectories are just files that keep track of other files, and, like any file, can grow if disk space permits.

The root directory may also contain files with names such as FILE0000.CHK and FILE0001.CHK— files created by CHKDSK and SCAN-DISK when they convert lost allocation units to files. These are clusters marked in the file allocation table (FAT) as belonging to a file, but which can't be traced back to a specific file's directory entry. Look at these files in a text editor such as DOS's EDIT.COM and see whether they contain any recoverable data. Otherwise, delete the FILE*.CHK files.

By the way, keep an eye out in the root for old copies of AUTOEXEC-.BAT and CONFIG.SYS, too. Many installation programs that modify these files make copies of the originals under a new name. Look for

different extensions, not just *.OLD or *.BAK. I've found versions of my AUTOEXEC.BAT and CONFIG.SYS with the extensions NU8, PCT, PCK, QWK, CR, and so on.

Everything in Its Place

Plan your directory structure so that it provides a logical place for every file. The "classic" structure is to have a subdirectory for each application, with a second-level subdirectory for data files: for example, C:\WP60 for Word-Perfect programs, C:\WP60\WP-DOCS for your documents, C:\WP60 \MACROS for macros, and so forth. Many installation programs will set up this directory structure for you.

An added benefit of keeping data separate from programs is that backing up is easier. Your data is what you want to back up, and often. After you back up program directories once, you generally don't have to do them again, because program files don't change. And, of course, you always have the option of rein-stalling the program from the original disks. Focusing only on data directories makes backing up easier and less time-consuming— and that means you're more likely to do it.

There are several standard sub-directories you'll want to set up and use for specific types of files. First and foremost is \DOS. Reserve this directory just for DOS's own files, such as FORMAT.COM and UNDELETE.EXE. Doing so makes it easier to upgrade to new DOS versions and lessens the possibility of losing non-DOS files.

Set up a \BAT or \BATCH directory for your batch files, and a \UTIL or \UTILITY directory for your utility programs, such as file viewers, archiving programs, and so on. This lets you keep

PATH BASICS

The PATH command tells DOS what subdirectories to search when it can't find the specified program in the current directory. DOS will search this "path" of directories before it returns its "Bad command or file name" error message. Some application programs also use the path to find files.

The path is specified by listing the directories to search, separated by semicolons:

```
PATH dirname1;dirname2;dirname3; . . . .
```

A typical path might look like this:

```
PATH C:\;C:\DOS;C:\UTIL;C:\WINDOWS;C:\BATCH;C:\DOS\NORTON
```

This path tells DOS to look in the following places (in the order listed) for the file specified, if DOS can't find it in the current directory:

C:\	root directory of drive C
C:\DOS	\DOS subdirectory on drive C
C:\UTIL	\UTIL subdirectory on drive C
C:\WINDOWS	\WINDOWS subdirectory on drive C
C:\BATCH	\BATCH subdirectory on drive C
C:\DOS\NORTON	\NORTON subdirectory within the \DOS subdirectory on drive C

Normally, you set the PATH in your AUTOEXEC.BAT file. If you have only a single hard drive, you can save a few characters by eliminating the C: drive identifier. The path above could be listed as follows:

```
PATH \;\DOS;\UTIL;\WINDOWS;\BATCH;\DOS\NORTON
```

You'll want to put your important subdirectories on the path, with the ones you access most often first, but don't include every subdirectory. Long paths mean that DOS must search each of the listed subdirectories, even when you make a typo or a spelling mistake.

There are a couple of ways of viewing the current path. If you simply enter PATH at the DOS prompt, DOS will display the path. You can also enter the SET command to see the path, because DOS keeps the path as an environment variable. The SET command will show all the environment variables, including PATH, PROMPT, and COMSPEC.

Because the path is an environment variable, you can access it and manipulate it in a batch file. That means you can change the path as needed, as this batch-file example shows:

```
@ECHO OFF
IF "%1"=="" GOTO NODIR
PATH=%PATH%;%1
ECHO The new path is %PATH%
GOTO END
:NODIR
ECHO Please enter the directory to add to the path
:END
```

When you run this batch file, the directory to add to the path is specified on the command line after the name of the batch file. If you omit it, %1 is blank and the IF statement true, so the program branches to the :NODIR label and displays an appropriate message. If %1 isn't blank, a new PATH statement is issued, with %1 appended to the current path (%PATH%); then the new path is displayed with the ECHO command.

—K.J.

DOS COMMANDS TO MANAGE DIRECTORIES

Several DOS commands can help you organize and manage directories on your hard drive:

CD or **CHDIR:** Changes the default directory. Special uses of the CD include:

CD	Shows the name of the current directory
**CD **	Moves you back to the root directory
CD ..	Moves you up one level to the "parent" directory of the directory in which you're currently working. The double period represents the parent directory of any subdirectory.

DELTREE: Deletes a directory and all its files and subdirectories, including hidden, system, and read-only files. DELTREE will prompt you for confirmation before deleting the branch, unless you include the /Y switch. You can't delete the current directory by typing DELTREE. You can delete all files in the current directory and all child subdirectories and their files, however, by including a file specification with the command (for example, DELTREE *.*).

DIR: Displays a list of files and subdirectories. In the standard format, directories are designated with a DIR in the file-size column. If you use the /W switch, directory names are shown in brackets—for example, [DOS]. Other useful switches are:

/S	Includes a listing for each subdirectory
/AD	Lists subdirectories only (no files)
/A-D	Lists files only (no subdirectories)
/OG	Lists subdirectories before files
/O-G	Lists subdirectories last

MD or **MKDIR:** Creates a new subdirectory. If you simply enter the directory name, the subdirectory is created below the current directory. You can indicate a different directory by including drive and path with the directory name.

MOVE: Can be used to rename, but not move, a subdirectory. Simply follow the existing directory name with the new name. You can't rename the current directory, and the new directory name can't duplicate the name of a file or subdirectory of the parent directory. And you can't specify a different path for the directory—that, of course, would literally be a move.

RD or **RMDIR:** Deletes a directory. The directory must be empty, containing no files or subdirectories. You can't remove the root directory, or the directory in which you're currently working (the default directory). RD is superseded in MS-DOS 6.x by DELTREE.

TREE: Displays a graphical "tree" of directories, showing their relationship in a visual way, much as the directory trees in the DOS Shell and Windows' File Manager do. If you include the /F switch, TREE will include the files in each directory.

—**K.J.**

everything together for easy access and editing. Include both directories in your PATH statement. (To learn how, see the sidebar "PATH Basics," page 25.) Also set up a \TEMP directory, and let DOS know about it by including this line in AUTOEXEC.BAT:

```
SET TEMP=C:\TEMP
```

DOS uses the \TEMP directory to store temporary files when doing input/output redirection ("piping"), and when task switching in the DOS Shell. Windows also uses this directory for temporary files while it's running. If you ever find files in this directory and neither the DOS Shell nor Windows is running, you can delete those temporary files. (If you have Windows, your \TEMP directory may be C:\WINDOWS\TEMP; in that case use SET TEMP=C:\WINDOWS \TEMP in your AUTOEXEC.BAT.)

You can also use the \TEMP directory as a temporary holding area for files you need only while you're currently working. Let's say you download a new utility in a ZIP file. Use the \TEMP directory to unzip the file. Look at the uncompressed files, then move the ones you need to the proper subdirectory (for instance, \UTILITY). Once that's done, you can delete the files in the \TEMP directory.

You can automate cleaning out the \TEMP directory in your AUTO-EXEC.BAT, by including these lines:

```
IF NOT EXIST C:\TEMP\*.* GOTO NEXT
ECHO Y | DEL C:\TEMP\*.*
:NEXT
```

The first line checks to see whether any files exist in the \TEMP directory. If so, they're deleted. The ECHO Y is piped to the DELETE command, automatically answering the "Are you sure?" question affirmatively.

Make Your MOVE

If you find yourself trying to figure out what a directory is from its

name, you can rename it to something more logical with MS-DOS 6.x's MOVE command. (For more on commands that help you get organized, see the sidebar "DOS Commands to Manage Directories," opposite.) MOVE can rename a directory, though you can't literally "move" it to another parent directory.

DOS 5 users can rename a directory through the DOS Shell. Highlight the directory from the directory tree, and select File/ Rename from the menu.

DOS 6.x's MOVE command also helps keep your disk organized by moving, rather than duplicating, files. COPY and XCOPY tend to jam directories with files, because many of us forget to go back and delete the originals.

If you're still running MS-DOS 5, you can create your own MOVE command with this DOSKEY macro:

```
DOSKEY MOVE=COPY $1 $2 $T
    IF EXIST $2\$1 DEL $1
```

This MOVE macro expects you to enter two parameters: the filename ($1) and the target location to which to copy ($2). The first command copies the file, and the second deletes the original file only if the copy was successful. Because the macro contains two separate DOS commands, they're separated by DOSKEY's $T symbol.

Clean Out Temporary, Backup, and Unused Files

Periodically go through and delete old backup (those with the BAK extension) and other temporary files. These files are probably scattered throughout your hard drive, just taking up space.

You can automate this process by outputting a directory listing to a batch file, then adding a DEL in front of each filename. (See the sidebar "Deleting a Disk's Worth of BAK Files," right.)

If you're a Windows user, look for files matching ~*.* and *.TMP. Windows uses these files while it's running, and normally deletes them when it shuts down. If Windows isn't running, you can safely delete these files. Just make sure you don't delete them if Windows is active.

Chances are your directories contain old files and utilities you don't use anymore. Besides cluttering up the directories, they're taking up precious space you can use for current files. You should move old files to floppy disks if you want to keep them, or delete them otherwise. Certainly retain those copies of your tax returns by archiving them to disk. But do you really need Christmas 1993's holiday letter?

Also, look carefully at your utility programs. Many of them are probably old and no longer necessary, especially if their functions have been incorporated into newer DOS versions.

How many of us still have utilities to read a keystroke from within a batch file, for example? DOS 6.x's CHOICE command makes those programs obsolete.

Similarly, an old system-report program may have been eclipsed by Microsoft Diagnostics (MSD), and a "smart copy" utility by the MOVE, XCOPY, and COPY overwrite warnings in MS-DOS 6.2.

Finally, move to disk all your ZIP, ARC, LZH, or other compressed files after you've uncompressed the individual files and moved them to the appropriate directories.

If you download files often from on-line services such as CompuServe or America Online, or from bulletin boards, you're likely to have many such compressed files on your hard-disk drive, wasting space. Archive those files to floppy disks and store them with your other "original" program disks.

Keeping House

Organizing your hard drive—and keeping it that way—isn't difficult, and will become a habit with a little practice.

Start now by creating a rational directory structure and storing files in logical places, and don't forget your housekeeping duties. Soon your directories won't be a maze anymore, but a clear path to the files you need. ■

DELETING A DISK'S WORTH OF BAK FILES

A good way to do disk housekeeping and free up disk space is to delete old BAK (backup) files. Many different programs create BAK files, which are probably scattered all over your hard drive. The DEL and ERASE commands don't have a way to delete across multiple subdirectories at one time. But with just a few commands you can create a batch file to delete all the BAK files in one fell swoop:

1. Create a file containing the names of all your BAK files as follows:

```
DIR C:\*.BAK /S /B > KILLBAK.BAT
```

This command displays a directory listing of all BAK files on drive C, starting at the root directory and including all subdirectories (the /S switch). The list is sent to a file called KILLBAK.BAT. The /B switch ensures that the files are listed with their full paths and filenames.

2. Edit the KILLBAK.BAT file, inserting DEL in front of each filename. If you're using DOS's EDIT.COM program, use the search-and-replace feature to replace each C:\ with DEL C:\. You might also want to include an @ECHO OFF as the first line in the file to suppress the display of the deletion messages as they execute.

3. Now run the KILLBAK.BAT file. It will go through and delete all BAK files in all your subdirectories.

—K.J.

SYSTEM SETUP

Without You I'm Nothing

Although you probably never even think about it, the information stored in your computer's CMOS is the most important data in your system.

by Hardin Brothers

Want to win a free drink at your local techno-bar? Just ask any self-styled expert to name the first information a computer reads when you turn it on. CONFIG.SYS and AUTOEXEC.BAT? Not even close! IO.SYS and MSDOS.SYS? Nope. The boot record from the hard disk? Closer, but no cigar.

The first thing your computer reads doesn't even have a filename. It's a collection of data held in CMOS (short for *complementary metal-oxide semiconductor*), a special memory chip that's kept alive by a small battery while the computer is turned off.

Without this information, your computer can't get very far. It can't boot up, it can't read from the floppy drives, and it can't find your hard drive. Whether you use DOS (with or without Windows), OS/2, Windows NT, a derivative of Unix, or any other operating system, you can't do anything until this data is recognized.

Technical Editor Hardin Brothers has been working with computers and writing about them for 15 years.

If for some reason you can't access the CMOS settings—if, say, your battery fails—your computer will simply sit there, running its fan and generating a little heat. You'll probably get a chilling error message on screen, saying something to the effect that all your CMOS data is lost.

In such circumstances, recovery is quick if you've taken one simple precautionary measure: documenting your CMOS settings. If you haven't, the road ahead is much rockier.

Few users take the time to look at the CMOS memory contents, and fewer still understand what CMOS does or how to protect it from corruption. Read on, and you too may avoid CMOS sorrow.

Searching for CMOS

Your CMOS information is kept in a block of CMOS RAM that doesn't have a CPU address. As a result, you can't use Debug's dump (D) command and view its contents. Unless you have special software, the only way you can read the CMOS contents is to run your computer's setup program.

And that can be a problem. On most computers with an 80286 CPU and some 386 machines, you have to run Setup from a disk file. Assume for a minute that you have Setup on your hard disk, and that your computer's battery fails. The CMOS memory will lose its information, and, whether you install a new battery or not, you must run Setup. Setup is on your hard disk, but you can't use your hard disk until you run Setup.

This classic Catch-22 has hit more than one user. The folly of putting Setup on hard disk has led to frantic calls to manufacturers, cross-country FedEx deliveries, and general annoyance all around. Also, Windows and many applications include a program called Setup, which copies files from a disk and prepares an application to run. If you're in the wrong directory when you run Setup, you might suddenly find yourself reinstalling (or even deinstalling) an important program.

So these days, most 386, 486, and Pentium computers store their setup programs in the computers' ROM chips. During bootup (or in

some cases at any time from the DOS prompt), you can press a key combination and run the program. For example, on my office computer I have to press Ctrl+Alt+Esc during bootup to run Setup.

The advantage of putting the setup program in ROM is that you can't lose it. As long as you can get your computer to boot up, you can run Setup. The disadvantage, which is slight, is that you can't run the computer's setup program except when you reboot. If you want to check the settings, you must close down all your applications, return to the DOS prompt, and then reboot.

If you have a built-in setup program, each time you boot up you'll probably see a message that tells you how to run it. If you have to run Setup from disk, copy it to a system floppy: Format a disk in drive A using the /s option, then copy Setup to the disk. Make sure you can boot and run Setup entirely from that disk.

Standard Settings

You won't find a CMOS setup on an 8088 computer, but if you have an 80286 or later system, I urge you to run Setup and become familiar with its screens and options. Someday you'll need it, probably when you face a deadline and feel least like messing with the inner workings of your computer.

While you're becoming familiar with Setup, copy down the information from its screens. If the program allows, use Print Screen or Shift+Print Screen to make a copy on your printer.

If you can't do that, get out a pencil and a piece of paper and spend 30 seconds copying down all the settings you see. Put your notes in a safe place, along with your emergency boot disk. Someday, your CMOS battery will fail. And when

it does, you'll either be thankful you have the information or curse your laziness for not copying it down.

When you run Setup, you'll see one or more screens of options and information (which vary with each computer). Some blocks of information are required for a computer to be "IBM compatible." For example, your setup screen should display the date and time. If they aren't correct, adjusting them is a good way to learn the subtleties of your computer's Setup interface.

Also, the setup screen lets you set the size and density of each floppy drive. Generally, modern computers can handle four types of floppies: 360K (5.25-inch double-density), 1.2MB (5.25-inch high-density), 720K (3.5-inch double-density), and 1.44MB (3.5-inch high-density). Choose the maximum capacity supported by each drive. Even if you set a drive at 1.44MB, for example, it will still be able to read and write lower-density disks.

Setup will also display, and let you change, information about your hard drives. To find the partition and boot records, your computer must know the size of your hard drives and how they're formatted.

Hard drives are usually designated by type. In the days of 10 and 20MB hard drives, the computer industry defined 45 or so standard "types." If you knew that your hard drive was type 5, for example, you could select that in any setup program and not have to worry about the details of cylinders, heads, sectors, and the other information Setup needs.

The computer world soon outgrew the standard hard-drive types. Manufacturers had to choose between updating a computer's BIOS (basic input/output system) chips every time a new and larger hard drive appeared or letting users and installers enter the drive speci-

fications manually. They wisely chose the latter course.

Today, you'll probably find about four dozen predefined hard-drive types listed on your Setup screen, plus two user-defined numbers. One of those user-defined numbers is for your first hard drive; the other is for the second. Hardly anyone uses the small, predefined drives on a modern computer. One consequence of user-defined drive settings is that you must record the type and each value displayed when you make a copy of your Setup screen.

You don't have to know what each field in the hard-drive specification means. If you're installing a new hard drive, you simply need the values for the drive. Often, those values are printed on a label attached to the hard-disk drive or included with the product's documentation. If you can't find them, ask your dealer or the manufacturer what values you should use.

You'll have to change only the floppy- and hard-disk settings if you change drives. Otherwise, as long as the battery stays charged and your computer doesn't forget what drives you have, these settings should never change. When the battery does fail and you have to reset the drive parameters, don't guess. If you set them incorrectly, you can make a hard drive inaccessible and even destroy the data it contains. Just pull out your copy of the settings and reenter them.

Extended Settings

The standard settings—date and time, floppy drives, and hard disks—were adopted by the computer industry as part of the IBM-compatible "standard." As far as I know, every 80286, 386, 486, and Pentium computer keeps that information in its CMOS memory and lets users adjust it with a setup program.

While industry standards set the minimum information that Setup should maintain, no standard exists for additional information. Every computer and BIOS manufacturer can decide what extra features to include in the Setup memory file and how those features should be organized.

Some settings are so common they're almost standard. Nearly every setup program keeps track of your primary video-display card, for example. On most computers, the display choices are MDA or monochrome, CGA, EGA, and VGA. The correct entry for almost all display cards manufactured in the last two or three years is VGA. Even monochrome displays are usually monochrome VGA, not the older MDA standard.

The computer uses information about the video type to display text; it assumes that graphics programs can determine by themselves the distinctions among types of VGA and Super VGA cards. If you upgrade from a CGA or EGA card to a VGA card, you'll have to change this entry. But if you change from one type of VGA card to another, even to an SVGA card, you don't have to change anything in Setup.

Memory Allocation

The second block of "almost standard" Setup data details the amount of memory installed in your computer and how that memory should be allocated. The Intel 386 and later CPUs give the computer a great deal of flexibility about how memory should be allocated, but DOS, Windows, and standard applications aren't as flexible.

Setup will probably keep track of the amount of base (DOS or real-mode) memory you have and the amount of extended memory. The former should almost always be set at 640K. The maximum for the amount of extended memory varies.

Before you set the amount of extended memory, you need to know how much RAM you have installed in your computer. You'll probably count your computer's RAM in megabytes (MB), but Setup will require that you specify amounts in kilobytes (K). One megabyte equals 1024K; you'll need to calculate the number of kilobytes of RAM you have before you adjust the memory settings.

You might think the amount of extended RAM in your computer should be the amount of total RAM minus the amount of base RAM. That would be true, except that most computers are set up to "shadow" the BIOS and, often, the video-card BIOS. The memory used for shadowing is neither extended nor base RAM, so it won't appear in either column of your Setup screen.

Shadowing may appear to steal memory from your computer, but it's a worthwhile investment. Shadowing moves the BIOS code from relatively slow ROM chips to much faster RAM chips. Because DOS, Windows, and virtually all applications make calls to routines in the BIOS, your computer will run noticeably faster if the BIOS is shadowed.

If you can't or don't establish shadowing in Setup, you can achieve the same effect by selecting the shadowing features of DOS's EMM386.EXE or a third-party memory manager.

But be careful that you don't enable shadowing of the same block of memory twice: once in the BIOS Setup program and a second time

---

CALLING IN A SUBSTITUTE

A deeply nested directory structure keeps your files organized, but makes it hard to get at the information you need. Imagine, for instance, that you want to check the payroll projections for the Katzen special project for the Midwest region for the first quarter of 1995 and that they're numbered consecutively from 01 to 73. To retrieve the first one, you'd need to work your way through your directory tree or type a sequence such as this one:

```
C:\MIDWEST\SPECPRO\KATZEN\PAYROLL\PROJ\1Q95
```

If you accessed those payroll files often, you'd find it worthwhile to use SUBST to create a fictitious drive for the subdirectory. For the example above, you can do that by typing the following at the DOS prompt:

```
SUBST drive: C:\MIDWEST\SPECPRO\KATZEN\PAYROLL\PROJ\1Q95
```

where *drive* is the letter of a drive not currently in use. For instance, if you have two floppy drives labeled A and B and two hard drives labeled C and D, you might substitute E for *drive*. (If your CONFIG.SYS file includes a LASTDRIVE setting, you may need to adjust it.) If you want the substitution to take effect at the start of every DOS session, add the command to your AUTOEXEC.BAT file instead.

A few words of caution are in order, however. Microsoft warns that problems may arise if you use SUBST with the following commands: ASSIGN, BACKUP, CHKDSK, DISKCOMP, DISKCOPY, FDISK, FORMAT, JOIN, LABEL, MIRROR, RECOVER, RESTORE, and SYS. You should also exercise care when using SUBST with CD, MD, and RD. Finally, don't run SUBST from a DOS session running under Windows; if you want to include SUBST commands in your AUTOEXEC.BAT, place them before the command that starts Windows. (The same precaution applies if you use another task-switching or multitasking environment.)

—Barry Colwell

with your memory manager. Your computer won't lock up, but you'll waste some of your extended memory.

Other Options

The number of additional features in Setup depends on the age of your computer and the manufacturer of its BIOS chips. Some computers offer no other Setup options, others feature three or four pages of options.

For example, my computer system, which has an Award BIOS, includes settings for the on-board serial and parallel ports, the motherboard's floppy-disk controller, and an optional high-speed memory cache. It also lets me set the speed of the memory chips I use, how I want the computer to handle errors, and what CPU speed I want to use each time I boot up.

I've seen other setup programs that set the keyboard typematic (key repeat) rate, run utility programs without loading DOS, and set bootup passwords that keep unauthorized users from even turning on the computer. Some turn on special BIOS calculators and hard-disk cache controllers, and switch the Num Lock state on or off during bootup.

Some computers display a custom message during bootup, let you configure the memory management chip set, and even set the sequence of drives the computer searches for an operating system when it starts.

You can adjust most of these settings to suit your computing habits, without endangering your computer. Adjusting the typematic rate, for example, is easy to do in the BIOS, with the DOS MODE command in your AUTOEXEC.BAT, or from the Keyboard icon in Windows' Control Panel.

All three adjustments will get you the same results; the advantage of using the BIOS is that you don't have to run any other software to speed up or slow down your keyboard.

DOS OPERATIONS AT YOUR FINGERTIPS

DOS in Windows TIP

When you're working in Windows, you can initiate a DOS operation, such as displaying a summary of memory use or listing the contents of a directory, by simply double-clicking on an icon. To make that possible, you must write a batch file to carry out the command, then create a program-information file (PIF) and an icon for the command.

For instance, suppose you want to execute the MEM /C command with a couple of mouse clicks. Using an editor that saves files in ASCII format, create the following two-line batch file, called MEM.BAT:

```
@ECHO OFF
MEM /C
```

Next, run PIF Editor, and in the Program Filename box, type the name of the batch file, preceding it with a pathname if necessary. Turn off the check box labeled Close Window on Exit, then exit PIF Editor and answer yes to Windows' "Save current changes?" message. Windows will save the file as MEM.PIF, place it in the \WINDOWS directory, and return you to Program Manager.

To create an icon and associate it with the PIF, highlight the group in which you want the icon to reside, and choose the File menu's New option. In the New Program Object dialog box, make sure that the Program Item option is selected, then select OK.

In the Program Item Properties dialog box, fill in a description (the text you want to appear below the icon: MEM /C) and the command line (the pathname of the PIF you created earlier: C:\WINDOWS \MEM.PIF). Select OK to save that information.

Now you're ready to try out the icon. When you double-click on it, a DOS window should open, and you'll see a summary of your system's memory use scroll down a DOS window. When it reaches the end of the summary, Windows changes the window's title bar from MEM /C to (INACTIVE MEM /C). That's Windows' way of telling you that the command isn't currently executing.

—Jack Nimersheim

My favorite setting, and one that my BIOS doesn't offer, lets you select which disk drives the computer will search for an operating system.

By default, most IBM-compatible computers search drive A (the first floppy drive) and then drive C (the first hard disk). When I wanted to install OS/2 on my machine as a second operating system, I could find it only on 3.5-inch disks. It had to be installed from drive A, but my 3.5-inch drive was drive B. I had to switch the cable and then alter the setup program so that my computer would see the 3.5-inch drive as drive A. It would have been so much easier to simply change an entry in Setup.

Explore with Caution

Don't be afraid to make changes in your setup program, but don't experiment blindly, either. If you don't know what a setting means, leave it alone until you read your computer's manual. Most manufacturers ship their computers with acceptable default settings in Setup. You might be able to make your computer faster or more efficient by changing some of them, and adapt it to your computing needs by changing some others.

But before you do anything at all, make sure you have a record of the current settings so that you can undo any changes you make and reconfigure your computer if the battery fails. ■

Working with Drives

A clever batch file makes program installation foolproof. Three simple steps help you manage the process better: Know what the kinds of drives you're working with, make sure they're ready, and always read the label!

by Robert L. Hummel

For me, one of the single greatest enhancements to personal computing has to be the standardization of installation programs. A few years ago, it would have been insane to try to install a new application without reading the instructions at least twice. Today, I open a new software package, root out disk number one, pop it into the drive, and type A:INSTALL or A:SETUP. More often than not, one of these commands will start the installation process. (Then I sit back and read the instructions.)

On the assumption that a good installation program can make even the simplest directions unnecessary, I decided to distribute one with my latest collection of utilities. But the typical professional installation program is a fancy executable that includes a graphics

Contributing Editor Robert L. Hummel is the author of PC Magazine Programmer's Technical Reference: The Processor and Coprocessor *and* PC Magazine Assembly Language Lab Notes *(both from Ziff-Davis Press, 1992).*

display, mouse support, and a decompression routine—leaving no room on the disk for my utilities! So I decided to use DOS's batch-file language to write a simple installation program. But I found that to make it elegant as well as simple, I had to construct some new tools first.

The Ground Rules

My first step was to establish these guidelines for my new installation batch program:

1. The program must not try to install itself on a floppy drive.
2. The program must not evoke an "Abort, Retry, Fail" message by attempting to read an empty floppy drive. It must prompt the user for corrective action.
3. The program must be able to detect when the correct disk is inserted into the source floppy drive. If not, it must prompt the user for corrective action.

Although these three criteria might seem like tall requirements for a batch program, it didn't take

me long to produce INSTALL.BAT. (See the first listing, page 34, top.) It uses a couple of batch tricks and no fewer than three of my new utilities to do its job.

Because the program's flow isn't strictly linear, we'll go through it a section at a time.

The first part of INSTALL.BAT is straightforward. A series of IF statements ensures that the three required arguments were specified on the command line. If not, the program displays the syntax of the command and then terminates.

For example, to install a program from a floppy disk in drive A to the directory D:\UTILS, you'd enter the following command:

```
INSTALL A: D: UTILS
```

In the next section, INSTALL tests for the presence of a fourth argument—even though the syntax specifies only three arguments. The reason for doing this will become obvious later in the discussion. For now, however, no fourth argument is present, so the IF test fails and execution continues with the next line.

Are You My Type?

DOS doesn't provide any commands to identify the type of device represented by a particular drive letter. In fact, one of DOS's major functions is to hide exactly that information from programs; DOS makes widely disparate hardware devices appear to have similar directory structures. That's why the DIR command works equally well on a floppy disk, a hard disk, a CD-ROM, or a RAM disk.

Sometimes, however, a program needs to tell what type of device underlies a drive letter. A batch program that backs up your drives would probably want to skip floppy, network, and CD-ROM drives, for example. Because DOS doesn't provide this capability, I wrote the utility DRVTYPE.COM.

Before you can use DRVTYPE, you must create it. To convert the Debug script DRVTYPE.SCR, shown in the second listing (page 34, bottom), to the executable program DRVTYPE.COM, follow the instructions in the "Debug Scripts" section of "How to Use This Magazine" (page 71 in this issue).

DRVTYPE accepts a single argument—a drive letter. If no drive is specified, DRVTYPE examines the current drive. (This lets your batch files identify the type of the default drive without knowing its drive letter.) DRVTYPE identifies the drive and returns its type as an exit code you can test with the ERRORLEVEL command. The first table (page 34, bottom) lists and defines the exit codes returned by DRVTYPE.

The next section of INSTALL.BAT uses DRVTYPE to verify that the target drive isn't a floppy by checking for a type code of 8 or higher. Similarly, it verifies that the source drive is a floppy.

A Duplication Trick

The next operation INSTALL.BAT performs is creating the destination directory on the target drive and copying itself there. DOS must read the batch program you're executing at least once for every line in the file. If you start executing a file in drive A, then switch floppies, DOS prompts you to replace the floppy containing the batch program—inconvenient and inelegant.

After copying itself to the hard disk, INSTALL then executes itself. The original three arguments are passed along with a fourth argument, the string HARDDRIVE. Execution of INSTALL thus begins all over at line 1.

The second time through INSTALL, the arguments are verified again. The section after the label :BEGIN is then executed. But this time, the IF test finds a fourth argument and fails; control transfers to the COPY1 label. INSTALL doesn't test the value of the fourth argument, only that it was present.

Technically, I could have located the IF test for the fourth argument closer to the top of program. That would have eliminated testing the three command-line arguments—%1, %2, and %3—the second time through. For clarity, though, it was better to position it after the block explaining the program's syntax.

Ready or Not?

The COPY1 section begins by echoing a message to put the first disk into the floppy drive. The PAUSE command then displays "Press any key to continue . . ." and waits for a keypress. Unfortunately, this is the point where most batch programs lose control of events. If you remove the floppy or swap disks unexpectedly, the program may become confused—or worse. INSTALL takes two steps to prevent this.

Before accessing the floppy drive, INSTALL uses the utility DRVREADY to verify that a floppy is in the drive and the door latched. Normally, trying to access an empty drive causes DOS to scrawl an "Abort, Retry, Fail?" message across the display. DRVREADY uses the BIOS to test the drive, avoiding DOS—and DOS's error message.

Create the DRVREADY.COM file from the Debug script DRVREADY.SCR, shown in the third listing (page 35, right). The second table (page 35, bottom right) lists and defines the exit codes returned by DRVTYPE.

INSTALL.BAT copies files from a floppy drive to a hard drive.

```
@ECHO OFF
::-------------------------
:: Sample installation program
:: copies files from two floppy disks
:: to a hard disk subdirectory.
::-------------------------
IF %1!==! GOTO USAGE
IF %2!==! GOTO USAGE
IF NOT %3!==! GOTO BEGIN

:USAGE
ECHO Syntax: INSTALL s: d: dir
ECHO    s:  = source floppy drive
ECHO    d:  = target hard drive
ECHO    dir = target directory
GOTO END

::-------------------------
:: If the program called itself, skip first section.
::-------------------------
:BEGIN
IF NOT %4!==! GOTO COPY1

::-------------------------
:: Verify that target d: is not a floppy drive.
::-------------------------
DRVTYPE %2
IF ERRORLEVEL 8 GOTO TARGETOK

ECHO Destination disk cannot be a floppy.
```

```
GOTO END
:TARGETOK

::-------------------------
:: Verify that the source drive is a floppy.
::-------------------------
DRVTYPE %1
IF ERRORLEVEL 1 IF NOT ERRORLEVEL 8 GOTO SOURCEOK

ECHO You must run this program from
ECHO Program Disk 1.
GOTO END
:SOURCEOK

::-------------------------
:: Create the destination directory
:: copy the install program there, and execute it
::-------------------------
MD %2\%3
COPY %0.BAT %2\%3
%2
CD \%3
%0 %1 %2 %3 HARDDRIVE

::-------------------------
:: Come here when running from a hard drive
::-------------------------
:COPY1
ECHO Put Program Disk 1 into drive %1
PAUSE
```

Continued opposite

DRVTYPE.SCR creates the DRVTYPE.COM program, which identifies the type of drive assigned to a logical drive letter.

```
N DRVTYPE.COM
A 100
JMP    12F
DW 0 0 0 0 0 0 0 0
DW 0 0 0 0 0 0 0 0
DB 0 0 1 3 2 6 7 B
DB 8 4 5 9 A
INC    AL
JZ     196
MOV    AH,52
INT    21
ES:
LES    BX,[BX+16]
MOV    CL,[5C]
OR     CL,CL
JNZ    14B
MOV    AH,19
INT    21
INC    AL
MOV    CL,AL
MOV    [102],CL
MOV    AH,58
MOV    AL,CL
DEC    AL
MUL    AH
ADD    BX,AX
SUB    AL,AL
ES:
MOV    CH,[BX+44]
AND    CH,C0
```

```
JZ     196
MOV    AL,D
CMP    CH,80
JZ     196
MOV    AL,E
CMP    CH,C0
JZ     196
MOV    AX,440D
SUB    BH,BH
MOV    BL,[102]
MOV    CX,860
MOV    DX,103
INT    21
MOV    AL,C
JB     196
MOV    AL,[104]
INC    AL
CMP    AL,A
JBE    192
MOV    AL,B
MOV    BX,123
XLAT
MOV    AH,4C
INT    21

RCX
9A
W
Q
```

End

DRVTYPE'S EXIT CODES

Type	Description
0	drive letter specified was invalid
1	320K/360K floppy
2	720K floppy
3	1.2MB floppy
4	1.44MB floppy
5	2.88MB floppy
6	8-inch single-density floppy
7	8-inch double-density floppy
8	tape drive
9	other block (disk) device
10	DOS can't identify the drive type
11	hard disk
12	RAM disk or drive created by INTERLNK
13	network drive
14	network drive or IFS (installable file system) device, such as a CD-ROM drive

Continued from p. 34 top

```
DRVREADY %1
IF ERRORLEVEL 1 GOTO COPY1

DRVLABEL %1 "UTILITIES#1"
IF ERRORLEVEL 1 GOTO COPY1

COPY A:*.*

::-------------------
:: Copy the second disk.
::-------------------
:COPY2
ECHO Put Program Disk 2 into drive %1.
PAUSE

DRVREADY %1
IF ERRORLEVEL 1 GOTO COPY2

DRVLABEL %1 "UTILITIES#2"
IF ERRORLEVEL 1 GOTO COPY2

COPY A:*.*

ECHO Installation complete.
DEL %0.BAT
:END
```

End

DRVLABEL.SCR creates DRVLABEL.COM, which uses exit codes to check for matching drive labels.

```
N DRVLABEL.COM          INC   SI
A 100                   LOOP  12D
CLD                     MOV   [SI],CL
OR    AL,AL             MOV   AX,BP
JZ    10B               STOSW
MOV   AL,FF             MOV   AL,5C
MOV   AH,4C             STOSB
INT   21                MOV   AH,4E
MOV   SI,81             MOV   CX,8
LODSW                   INT   21
DEC   SI                MOV   AL,1
CMP   AH,3A             JB    107
JNZ   10E               DEC   AL
MOV   BP,AX             JMP   107
LODSB
CMP   AL,D              RCX
JZ    105               49
CMP   AL,22             W
JNZ   117               Q
LEA   DI,[SI-3]
MOV   DX,DI
ADD   SI,8
MOV   AL,2E
MOV   CX,4
XCHG  AL,[SI]                        End
```

DRVREADY.SCR creates DRVREADY.COM, used by INSTALL.BAT to verify that a floppy is in the drive and that the door is latched.

```
N DRVREADY.COM
A 100
OR    AL,AL
JZ    108
MOV   AH,4C
INT   21
MOV   DL,[5C]
DEC   DL
JNS   116
DEC   DL
MOV   AL,DL
JMP   104
MOV   CL,3
PUSH  CX
MOV   AH,4
MOV   AL,1
MOV   CH,0
MOV   CL,1
MOV   DH,0
INT   13
POP   CX
JNB   132
LOOP  118
SUB   AH,AH
INT   13
MOV   AL,1
JMP   104
SUB   AL,AL
JMP   104

RCX
36
W
Q
```

End

DRVREADY'S EXIT CODES

Type	Description
0	drive is ready
1	drive is not ready
254	no drive was specified on the command line
255	specified drive is invalid

Hard drives, by their nature, are always ready. Other types of drives, such as network and CD-ROM drives, aren't controlled by the BIOS. So before using DRVREADY in your own programs, use DRVTYPE to make sure that the drive is a floppy.

Look for the Label

A program that installs itself from a single floppy disk can verify that the drive is ready and then proceed. Programs that use more than one source floppy, such as INSTALL-.BAT, must perform some extra work to make sure you've inserted the correct floppy in the correct sequence.

As you may recall, a disk's label (also called its *volume name*) is an electronic signature created with the LABEL command or at the time you formatted the disk. You can display a drive's label, if it has one, with the DIR command. The label can include a maximum of 11 alpha-numeric characters. It can't contain any of the following special DOS characters:

```
* ? " / \ | . , ; : +
= [ ] ( ) & ^ < >
```

You can give your source disks unique labels, such as DISK 1, DISK 2, and so on. But until your batch programs can read the labels, something DOS doesn't support, labels won't do you much good. My DRVLABEL utility addresses that shortcoming. (For another approach to the problem, see "Turn Up the Volume," by Hardin Brothers, *DOS World* #17, September 1994, page 34.)

Create the DRVLABEL.COM file from the Debug script DRVLABEL.SCR, shown in the fourth listing (page 35, bottom left). The syntax for DRVLABEL is as follows:

```
DRVLABEL d: "drive-label"
```

where *d*: is the drive to be examined and "*drive-label*" is the label that must be matched. DRVLABEL returns 0 if the label matches, 1 if it doesn't, and 255 if an error occurs.

Note that the label must appear in quotes and must be exactly 11 characters long. If necessary, pad the label on the right with blanks. The label argument isn't case sensitive. Thus the label arguments "12345678901" and "My Big Disk" are valid. The arguments "Disk1" and MAINDISK are invalid. (The first is too short; the second lacks quotes.) Before accessing each disk, INSTALL checks for the correct volume label.

Putting It Together

Building smarter installation programs isn't hard if you've got the right tools. The utilities presented here can help you accomplish that. But don't stop there. Once you get used to them, I'm sure that DRVTYPE, DRVREADY, and DRVLABEL will find countless other uses in your everyday batch programs—just as they do in mine. ■

DOS TIP

COMING OUT RIGHT IN THE END

I recently wrote a batch file that had to exit from the middle of itself rather than at the end. To do that, I created an empty batch file, called END.BAT, which runs from the main batch file. Because END.BAT doesn't contain any commands, it simply exits to DOS. Because it's zero bytes long, it doesn't take up any room on my disk.

The trouble is, I sometimes pass my batch files along to other people. If someone who doesn't have END.BAT tries to use my program, he or she will get a "File not found" error message, and the batch file will continue executing. To solve this problem, I wrote a short routine that uses the command REM>C:\END.BAT to create END.BAT and then stores the file in the root directory.

My demonstration batch file, TEST.BAT, shows how the routine works. If you type TEST at the DOS prompt and follow it with a parameter, such as CONTINUE, the batch file jumps to the CONTINUE subroutine, where you place the commands composing the main program. If you type TEST without including a parameter, as a user who hasn't run the program might, the subroutine called END executes.

—Eric Maloney

TEST.BAT is a demonstration program that shows how you may use a routine to create the empty file END.BAT the first time a batch file runs.

```
@ECHO OFF
IF "%1"=="CONTINUE" GOTO CONTINUE
IF "%1"=="continue" GOTO CONTINUE
:END
ECHO You typed TEST without a parameter. The batch file will
ECHO now create an empty file called END.BAT, execute it, and
ECHO return you to DOS. From now on, type TEST plus a parameter
ECHO (TEST CONTINUE, for example) whenever you execute TEST.BAT.
REM>C:\END.BAT
C:\END
:CONTINUE
REM When you type TEST CONTINUE (or TEST followed by another
REM parameter), the main batch file jumps to the subroutine
REM called CONTINUE and the program continues to execute.
REM Place the rest of your batch file here.
```
End

B A T C H - F I L E   M E D I C

PROMPT Delivery: Capturing Data in DOS

By grabbing information with the PROMPT command, your batch files can perform tasks on demand.

by Hardin Brothers

Many of us would like our AUTOEXEC.BAT files to perform different activities on different days—a weekly backup every Friday, for example. Such a once-a-day batch file is difficult to write without resorting to a Debug script.

I found a better way recently while playing around with the DOS PROMPT command. You probably use PROMPT to change the appearance of your command-line prompt. But you can also use PROMPT to deliver the current day to a batch file. Once you've got the day in an environment variable, you can perform actions based on what day it is.

The trick is to reset the prompt temporarily to display the day and capture its output. You can use the technique to grab other information the PROMPT command can display, as well, including text strings, current drive (without current path), date, time, and DOS version.

Capturing PROMPT information wasn't quite as easy as I thought it

Technical Editor Hardin Brothers has been working with computers and writing about them for 15 years.

would be, but eventually I developed a nearly painless method. I'll explain it and show you some batch-file fragments you can add to your own batch files. I'll end with a simple and powerful once-a-day program written entirely with DOS's batch language.

If at First . . .

Before I could do anything else, I had to find a way to capture the DOS prompt in a file. My first attempt was a short batch file that looked like this:

```
@ECHO OFF
SET OldPrompt=%Prompt%
PROMPT Testing > TEST.FIL
PROMPT %OldPrompt%
SET OldPrompt=
```

I wanted to reset the DOS prompt to the word *Testing* and then save that word to a file called TEST.FIL. I ran my short program, then looked excitedly at the contents of TEST.FIL. It was empty! Chalk up one failure.

It didn't take me long to figure out what had gone wrong. The redirection in the second line below @ECHO OFF sent the output of the

PROMPT command to TEST.FIL. But PROMPT doesn't have any direct output. It simply changes the DOS prompt, and the results appear when the prompt is displayed next.

Waiting for Another Idea

Okay, we all make mistakes. But I was baffled: I had this great idea for capturing essential information, but it wouldn't quite work.

As far as I know, there's no way to echo the DOS prompt to a file, because DOS doesn't interpret the prompt itself as a command. But there had to be a way.

Wait a minute. COMMAND is a command; it runs a new copy of COMMAND.COM, the DOS command interpreter. And COMMAND.COM has output—the DOS prompt. Maybe I could redirect COMMAND.COM to a file. But if I did redirect its output to a file, how would I regain control of the computer? I wouldn't be able to type EXIT and get back to my original prompt.

Well, COMMAND /C executes one command and exits automatically. So my second attempt was this:

```
COMMAND /C > TEST.FIL
```

I got back control of my computer, as I expected, but once again I had an empty file. Either the new copy of COMMAND.COM was exiting before it displayed a prompt or this idea wouldn't work either.

To be sure that my second idea wasn't a complete failure, I created a one-line file called TEST.BAT:

```
@ABCDEFGH
```

Normally, if you type ABCDEFGH at the command line, DOS displays its infamous "Bad command or file name" message. So if I ran TEST.BAT nothing would happen, but DOS would display the prompt.

After I created that batch file, I was ready for my final test. I typed this line at the DOS prompt:

```
COMMAND /C TEST.BAT > TEST.FIL
```

After a short pause, I saw "Bad command or file name" and then my regular prompt. And this time, TEST.FIL wasn't empty: It contained my standard DOS prompt. Success! (Almost.)

What Happened?

To understand both the on-screen message and the contents of the test file, you need a little background on the way DOS works. Whenever any program, including COMMAND.COM, starts, DOS gives it five channels of communication. The three most important are *standard input* (usually the keyboard), *standard output* (usually the screen), and *standard error* (also the screen). When you use redirection in a batch file (<, >, and >>), you send a program the contents of a file as standard input or you send standard output to a file.

So my test line (COMMAND /C TEST.BAT > TEST.FIL) was doing several things. First, it ran a new copy of COMMAND.COM. The /C TEST.BAT portion of the line told that copy of COMMAND.COM to execute TEST.BAT and then exit. And > TEST.FIL redirected standard output to a file called TEST.FIL.

While the new copy of COMMAND.COM was running the batch file, it sent the DOS prompt to standard output, as it tried to execute my nonsense command, ABCDEFGH. That accounted for the copy of my DOS prompt that appeared in the output file. And when it found that it couldn't execute my command, it sent the bad-command message to standard error, which is why that line appeared on screen.

So far, so good. I'd successfully captured a copy of the DOS prompt in a file, which was my original goal. And I managed to do it without losing control of the computer and having to reboot.

Doing Something Useful

With the technical stuff out of the way, it was time to get down to work. Let's start with a simple task: capturing the current drive letter and storing it in an environment variable called DRIVE.

Here's an outline of one way to solve that problem:

1. Create a prompt that displays the current drive by using $N as part of the prompt.
2. Capture that prompt in a file.
3. Move the drive letter from the file to an environment variable.

We've already solved the first two steps. And step 3 is just a variation of a problem we've faced many times in this column in the past.

The first of the accompanying listings, DRIVE1.BAT (opposite, top), contains one somewhat clumsy method of using the PROMPT command to capture the current drive letter. (These programs assume that all your BAT files are in C:\BAT; if you're using a different subdirectory, substitute that name where appropriate.) The program starts in line 10, which creates a new

THE LOWDOWN ON LOADING HIGH

DOS 5 and later give you the tools to do something productive with the upper memory area (UMA), the 384K of empty space between the top of conventional memory (640K) and the start of extended memory (1,024K). After loading EMM386.EXE, DOS's UMA manager, you can use the commands DEVICEHIGH and LOADHIGH to stuff into the UMA programs that normally would occupy memory below the 640K mark—memory that's better used to run applications. In addition, DOS itself may take advantage of the high memory area (HMA), the 64K area of memory starting at 1024K. By loading a portion of itself there, DOS frees up even more conventional memory.

But, on some systems, after you load DOS, device drivers, and memory-resident programs into upper memory, the first free byte of memory is below the 64K mark. Many older applications, created when DOS always took up at least 64K, can't handle this situation, and you receive a "Packed file corrupt" message. Sometimes your system even locks up.

To address this problem, DOS includes LOADFIX.EXE, a miniature program launcher. When you run LOADFIX, it first allocates all memory below 64K, then runs the specified application. When the application exits, so does LOADFIX, freeing all the memory it occupied. The syntax for LOADFIX is as follows:

```
LOADFIX [d:][path]program
```

where *program* is the name of the application program and *d* and *path* are the drive and path where the program resides.

—Robert L. Hummel

Line 12 starts a new copy of COMMAND.COM and sends it DRIVE-CMD.BAT as its one and only command. The same line takes the output of this command (one copy of the new prompt) and places it in a file called DRIVETMP.BAT.

Now we have three temporary one-line batch files. Line 13 starts to make sense of all this by calling DRIVETMP.BAT. That batch file, in turn, runs DOIT.BAT and sends it the drive letter as a command-line parameter. And DOIT creates the parameter in the environment before it returns control to DRIVE1-.BAT. Lines 15 through 17 delete the temporary batch files; line 19 displays the results.

There's only one last detail to understand. If DRIVECMD.BAT changes the prompt, why doesn't the new DOS prompt appear on screen when everything's done?

COMMAND.COM stores the prompt string in the environment. (You can see it by typing SET at the DOS prompt.) When you start a new copy of COMMAND.COM in line 12, it inherits a copy of the environment. It changes that copy to create the new prompt, but when that copy of COMMAND.COM ends, its temporary copy of the environment disappears. The original version of COMMAND.COM regains control with its original copy of the environ-

---

DRIVE1.BAT. This batch file captures the DOS prompt and saves the current drive letter in the environment. Note that lines are numbered for your reference only. Don't type in the line numbers or the colons following them; start with the command in each line.

```
1 : @ECHO OFF
2 : REM  -- DRIVE1.BAT --
3 : REM  This is one way to capture the
4 : REM  DOS prompt and save the current
5 : REM  drive letter in the environment.
6 : REM  This assumes that you have a
7 : REM  directory called C:\BAT.
8 :
9 :
10 : ECHO @SET DRIVE=%%1  > C:\BAT\DOIT.BAT
11 : ECHO @PROMPT DOIT $N > C:\BAT\DRIVECMD.BAT
12 : COMMAND /C C:\BAT\DRIVECMD.BAT > C:\BAT\DRIVETMP.BAT
13 : CALL C:\BAT\DRIVETMP.BAT
14 :
15 : DEL C:\BAT\DOIT.BAT > NUL
16 : DEL C:\BAT\DRIVECMD.BAT > NUL
17 : DEL C:\BAT\DRIVETMP.BAT > NUL
18 :
19 : ECHO The current drive is %DRIVE%
```
End

batch file (DOIT.BAT) containing just one line:

```
@SET DRIVE=%1
```

The listing uses two % signs because one will be lost during execution of the ECHO command. This batch file's job will be to receive the drive letter as a command-line parameter and place it into the environment.

Line 11 creates a second batch file, called DRIVECMD.BAT. This

batch file also contains one line, as follows:

```
@PROMPT DOIT $N
```

This line will create a new prompt. The first word on the line, DOIT, is the command that will eventually start DOIT.BAT. The $N is the PROMPT parameter for the letter of the current drive. For example, if you run this set of batch files from drive D, the prompt created by this line will be DOIT D.

GETINFO.BAT. This batch file collects information from the DOS prompt and places it in the environment. Note that lines are numbered for your reference only. Don't type in the line numbers or the colons following them; start with the command in each line.

```
1 : @ECHO OFF
2 :
3 : REM  -- GETINFO.BAT --
4 : REM  Collects information from the
5 : REM  the DOS prompt and places it
6 : REM  in the environment.
7 :
8 : IF NOT "%1"=="" GOTO SaveAll
9 : ECHO @PROMPT %0 $N $P $D $T $V >
        C:\BAT\GETINTMP.BAT
10 : COMMAND /C C:\BAT\GETINTMP.BAT >
        C:\BAT\GETINTP2.BAT
11 : C:\BAT\GETINTP2.BAT
12 :
13 : REM  You get here the second time this
14 : REM  runs, with the information in
15 : REM  the command-line parameters.
16 :
17 : :SaveAll
18 : DEL C:\BAT\GETINTMP.BAT > NUL
19 : DEL C:\BAT\GETINTP2.BAT > NUL
20 : SET DRIVE=%1
21 : SET CURPATH=%2
22 : SET DAY=%3
23 : SET DATE=%4
24 : SET TIME=%5
25 : SET DOSVER=%8
26 :
27 : ECHO Drive is %DRIVE%
28 : ECHO Current path is %CURPATH%
29 : ECHO The day is %DAY%
30 : ECHO The date is %DATE%
31 : ECHO The time is %TIME%
32 : ECHO DOS version is %DOSVER%
```
End

ment, including its prompt. As far as it knows, nothing has changed.

A Little Fancier

The second listing, GETINFO.BAT (page 39, bottom), shows all the information DOS can extract from the PROMPT command. It creates two temporary batch files, not three, because the instructions to copy information from the command line to the environment are in the main batch file. Otherwise, it's identical in concept to the first listing.

Line 9 creates a longer prompt than the one in the first program. It replaces %0 with the name of the running batch file, in this case GET-

INFO.BAT. Using %0 guarantees that the batch file will continue to work even if you change its name. $N is the PROMPT command for the current drive, $P is the current path, $D is both the day and the date, $T is the time, and $V is the DOS version.

There's only one other unusual aspect to this program. If you look at lines 24 and 25 carefully, you'll see that command-line parameters %6 and %7 are completely ignored. If you set your prompt temporarily to $V, you'll see why. Instead of just inserting the version number into the prompt, $V inserts something like this message:

MS-DOS Version 6.20

The program gets rid of *MS-DOS Version* by ignoring %6 and %7, the command-line parameters containing those words. (By the way, for *IBM DOS*, use %9 in line 25 instead of %8.)

Once you've tested the second program, remove the ECHO statements from lines 27 to 32 and save the program as GETINFO.BAT.

Once-a-Day

We'll end this month with a version of ONCEADAY.BAT (third listing, left). We've published several other versions of this program in past issues of *DOS World*; many of them depend on Debug scripts to capture the date. This time, we'll use GETINFO.BAT, from the second listing, to get the date. We'll also use ONCEADAY.BAT to start a special activity each Friday. (You could use it to run your weekly backups, for example.)

The program begins by calling GETINFO to load information about everything, including the date, into the environment. Then it uses the date to create a filename. If that file already exists, the once-a-day activity has already been performed: This isn't the first time you've booted your computer today.

If the special file doesn't exist, you haven't booted yet today. In that case, the program performs whatever activities you specify for daily execution.

In addition, the program checks to see whether it's Friday. If so, it will perform weekly activities on the first Friday boot-up. Of course, you'll have to write the commands you want DOS to execute once a day or once a week.

We can make ONCEADAY.BAT fancier, letting it execute some activity every other day or every third day, for example. But that's a task that will have to wait for a future column. Besides, you'll need time to incorporate this month's ideas into your own batch files, won't you? ■

ONCEADAY.BAT. This batch file uses information from GETINFO.BAT to decide whether you're calling the program for the first time today; if so, it performs daily housekeeping routines. Note that lines are numbered for your reference only. Don't type in the line numbers or the colons following them; start with the command in each line.

```
 1 : @ECHO OFF
 2 :
 3 : REM -- ONCEADAY.BAT --
 4 : REM This program uses information from
 5 : REM GETINFO.BAT to decide whether this
 6 : REM is the first time you've called
 7 : REM ONCEADAY today. If so, this
 8 : REM program performs whatever daily
 9 : REM housekeeping routines you need.
10 :
11 : CALL C:\BAT\GETINFO.BAT
12 : IF "%DATE%"=="" GOTO Failed
13 : IF NOT EXIST C:\BAT\%DATE%.DAY GOTO FirstTime
14 :
15 : ECHO Your daily activity has already been performed.
16 : GOTO End
17 :
18 : :FirstTime
19 :
20 : REM Get rid of old *.DAY files and
21 : REM create a new, zero-byte file.
22 :
23 : DEL C:\BAT\*.DAY > NUL
24 : REM > C:\BAT\%DATE%.DAY
25 : ECHO Performing once-a-day activities.
26 : REM Place activity commands here.
27 :
28 : IF NOT "%DAY%"=="Fri" GOTO End
29 : ECHO TGIF!  It's Friday!
30 : REM Place weekly activities here.
31 : GOTO End
32 :
33 : :Failed
34 : ECHO GETINFO failed to place the date
35 : ECHO in the environment.
36 :
37 : :End
```

End

MULTIMEDIA UPGRADES

DOS and Multimedia: You Can Do It!

Who says DOS users have missed the multimedia bandwagon?
A few simple upgrades can give you access to many exciting programs.

by Michael Nadeau

Many great multimedia game, educational, and reference CD-ROMs are available for DOS PCs. In part, you have Windows to thank for it. Graphics performance under Windows is notoriously slow, and publishers know their graphics-based software will look better running under DOS.

But is your DOS PC multimedia-ready? Depending on your setup, you'll be faced with three alternatives: a quick upgrade and a little tweaking of your system files, purchasing a multimedia kit, or buying a whole new system. To find out where you fit in and what to do, follow the steps discussed below.

Upgrade or Start Over?
Only Your CPU Knows for Sure

Even under DOS, multimedia software places great demands on your PC. It needs a fast processor, more RAM, and more space on your hard drive. The first of the accompanying tables ("DOS Multimedia Require-

Michael Nadeau is the author of The BYTE Guide to CD-ROM: Second Edition *(Osborne/ McGraw-Hill, 1995). You can reach him via the Internet at* **miken@bix.com**.

ments," page 43, bottom) lists minimum and recommended features.

To begin evaluating your system, check the owner's manual to find out your PC's processor type; for multimedia, you'll need at least a 25MHz 386 CPU (central processing unit). More and more titles, however, demand at least a 25MHz 486-class CPU. Processor upgrades are available for many 286 and later systems, but they're expensive (see the second table, "Upgrade Costs," page 43, top) and won't match the performance of a system designed for the faster CPU. When considering a CPU upgrade, also check the condition of your hard drive, keyboard, and other components that are susceptible to wear.

Bottom line: Rather than throw good money after bad in an attempt to upgrade the CPU on an old 286 or 386, buy a new, ready-to-go multimedia system. Some fully outfitted brand-name systems are available through mail-order catalogues for as little as $1200.

A Memory Quick-Fix

If your CPU is up to speed, then memory is a simpler matter when considering a system upgrade. With DOS, you can get by with 4MB, though many multimedia applications are happier with 8MB. If you need to add RAM, your PC owner's manual should tell you what type of memory your computer uses and how to install it.

A hard disk for multimedia applications should be, at the very least, an uncrowded 100MB drive, but 160MB or more is preferable. Most multimedia CD-ROMs load files onto your hard drive to take advantage of its faster access rate. Usually, the files take only a megabyte or two, but some—particularly games—can place 50MB or more of files on your hard disk.

If you don't want to upgrade to a large-capacity hard drive, you can use a disk-compression utility such as Multimedia Stacker from Stac Electronics (see the review, "Turning Less Into More," page 42) or MS-DOS 6.x's DriveSpace to create room by "shrinking" the size of the software files on your existing hard drive. But be prepared to trade off speed for space, and the reality that some multimedia applications won't run on a compressed drive.

TURNING LESS INTO MORE

Any multimedia PC that's running disk-compression software, SCSI device drivers, CD-ROM extensions, and other TSR utilities will suffer a significant drop in conventional memory. You may even discover that multimedia packages and games can't start in the 450K to 520K available after bootup. Even if Smart-Drive and your disk-compression drivers load above 640K, there's no room left in upper memory for SCSI drivers and CD-ROM extensions.

Enter Stac Electronics' **Multimedia Stacker**, a package featuring two trustworthy products from separate vendors: Helix's Multimedia Cloaking Utilities, designed to help you win the battle against too little memory to run your multimedia programs; and Stacker 4.0 for DOS and Windows, considered the most efficient DOS disk-compression utility on the market.

Both products will cooperate with any DOS memory manager, including Microsoft's basic HIMEM.SYS/EMM386 combination. Note that although the two products are bundled under the single title Multimedia Stacker, they require separate installation. Stacker can use either Stac's DPMS.EXE or Helix's CLOAKING.EXE to move program code into extended memory. These separate installations place both drivers into CONFIG.SYS; you can grab an additional 5K by commenting out the DEVICE= DPMS.EXE line if DEVICE= CLOAKING.EXE is present.

Stacker lets you transparently expand the storage space on your hard disk by as much as 250 percent. The software accomplishes this by packing all files from a specified partition into a special compressed archive. Stacker will produce an average of 20 percent more available disk space than the DoubleSpace/DriveSpace compression programs Microsoft bundles with MS-DOS. Stacker is also more flexible than the Microsoft compression utilities, since it lets you compress a RAM disk or create portable compressed floppy disks that can be read even on systems where Stacker isn't installed. You can run its easy interface from either DOS or Windows, and its disk optimizer is faster than Microsoft's Defrag.

Helix's Multimedia Cloaking Utilities free up an additional 64K to 96K of conventional memory by replacing the Microsoft disk cache, mouse driver, and CD-ROM extension with drivers that load above 1MB in extended memory. CacheCloak includes all the features of SmartDrive's disk cache, leaving a conventional-memory stub of less than 5K (no matter how large you make the cache). CacheCloak can also handle multiple CD-ROM drives. Helix's MSCDEX substitute duplicates all Microsoft features, and also lets you assign specific drive letters to multiple CD-ROM drive units. The Helix mouse driver was designed by Logitech and is fully compatible with all Microsoft and Logitech mouse types. MouseCloak also includes a series of advanced customization features that control mouse speed in various screen resolutions. In tests with Quarterdeck's QEMM memory manager, Cloaking Utilities moved 89K of resident drivers and TSRs out of the first megabyte and into extended memory. (If you use Helix's Netroom as your memory manager, you may get a slightly smaller memory gain, since Netroom already uses Cloaking technology and the CacheCloak disk.)

In addition to Stac's DOS and Windows front ends for Stacker, the Multimedia package installs Discover, a memory-inventory utility I found much more complete and useful than Microsoft's MEM command.

Multimedia Stacker, $99.95; requires 386 PC or later, minimum 1MB RAM, DOS 5 or later, Windows 3.1 or later; Stac Electronics, 12636 High Bluff Drive, San Diego, CA 92130, 619-929-3900.

—**Lenny Bailes**

Sound Decisions

If your PC doesn't have a sound card, choose a 16-bit board that can produce *waveform audio* (also known simply as *wave*), because its faster sampling rates capture more of your multimedia application's original sound. Wave is digital sound sampled at 11KHz to 44KHz; professional quality is considered to be 16-bit 44KHz sound. An 8-bit board is incapable of such performance.

Creative Labs' Sound Blaster is the dominant industry standard, and most popular sound cards emulate its format. Other significant formats include Ad Lib, MediaVision's Pro Audio, and Windows. Look for a card that supports multiple standards. (See the sidebar "Unraveling the Multimedia Mystery," page 44, for a list of current cards.)

To install a sound card, plug it into an open 16-bit expansion slot inside your PC, run the setup program that came with it, and follow the on-screen prompts.

Adding a sound card is just the first step. You'll also need headphones or speakers, which you plug into the audio jacks of your sound card, at the back of your PC. Jacks on most sound cards, and on the front of CD-ROM drives, accept headphones commonly used with portable stereos. You can also use a converter to plug in the larger headphones used with home stereos.

PC speakers come in a wide range of price and quality, but keep in mind that lower-quality speakers won't adequately live up to multimedia's 16-bit sound. Speakers come with built-in amplifiers; the higher the wattage, the louder the volume. Some speakers include a battery option to boost power, while more expensive models add a subwoofer: a separate speaker dedicated to boosting the bass range, for a richer sound.

The Drive for Success

When selecting a CD-ROM drive for multimedia, you'll need to choose between double-speed and quad-

speed. Note that although a quad-speed drive can access data from a CD-ROM roughly twice as fast as a double-speed unit (600K versus 300K per second), the rate at which the rest of your computer can process that data remains unchanged.

Furthermore, most video used in multimedia CD-ROMs is recorded at 150K per second, and a quad-speed drive can read that video no faster than that. So, unless you have a very fast processor (66MHz 486 or better) and a fast 32-bit graphics accelerator card, you'll see little improvement in throughput with a twice-as-expensive quad-speed drive.

If you don't want to buy a new system, but you have neither a CD-ROM drive nor a sound board, your best bet is to buy a multimedia upgrade kit rather than put an upgrade together piecemeal. To learn more about choosing a kit, see the sidebars, "Unraveling the Multimedia Mystery" (page 44) and "Kit Caveats . . ." (page 45).

Finally, consider your power supply, which regulates the power coming from the AC outlet to the various components of your computer. Adding hardware means that your power supply must provide more juice. If it lacks the capacity, certain components may malfunction or the power supply may fail completely.

As a rule of thumb, multimedia PCs should have at least a 150-watt power supply. Check the mail-order advertisements in popular computer magazines for suppliers. Before ordering, make sure that the power supply will fit your system and won't require rewiring the line voltage.

Go Configure

No upgrade is complete without experiencing the joy of modifying your DOS system files. CD-ROM drives and sound cards should come with set-up software that installs the necessary drivers and makes the appropriate changes in your CONFIG.SYS and AUTOEXEC.BAT files. But because of the demands multimedia software will make on your system, you'll probably need to make further modifications.

First, the new multimedia drivers you install will consume precious system memory, leaving less room for DOS and your applications. Once you have everything installed, type MEM at the DOS prompt. Some multimedia applications require at least 600K of conventional memory; if the "largest executable program size" shows less than that available, you'll need to move something into high memory.

If you use DOS 6.x, just run the MemMaker utility; type MEMMAKER at the DOS prompt and follow the on-screen instructions. MemMaker will load the new drivers into high memory. If you're running DOS 5, load the operating system into high memory with the following lines in CONFIG.SYS:

```
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH
```

You can also use the LOADHIGH (or LH) command to load some device drivers into high memory, but check your DOS manual for instructions. Alternatively, you can use a third-party memory-management utility, such as QEMM, from Quarterdeck Office Systems.

Many DOS multimedia applications require extended memory, typically 2MB. You might have to disable SmartDrive or any RAM-disk utilities you have running to free up enough extended memory. Just type REM or a semicolon in front of the appropriate lines in the CONFIG.SYS file. To be safe, you might want to keep two versions of CONFIG.SYS on your hard drive: your standard one and another that frees up the maximum amount of extended memory.

Sometimes, a DOS multimedia application will refuse to install itself from the CD-ROM. To resolve the situation, the documentation usually tells you to create a boot

UPGRADE COSTS

Component	Price Range
66MHz 486DX2	$225–$300
100MHz 486DX4	$500–$550
1MB RAM	$40–$60
200MB hard drive (IDE)	$160–$200
500MB hard drive (IDE)	$220–$300
double-speed CD-ROM drive	$100–$175
quad-speed CD-ROM drive	$200–$400
16-bit sound card	$100–$300
multimedia upgrade kit	$240–$600
25MHz 486SX multimedia PC (fully equipped)	$1200–$1500

DOS MULTIMEDIA REQUIREMENTS

In addition to these requirements, make sure you have an open bay for the CD-ROM drive and enough open expansion slots for a sound board or CD-ROM drive interface. Check the label on the power supply for its specifications, and while you're at it, see whether your PC has a CPU upgrade slot. (Note: SB = Sound Blaster.)

	Minimum	Recommended
CPU	25MHz 386	33MHz 486
RAM	4MB	8MB
hard disk	100MB	160MB
CD-ROM drive	double-speed	double-speed
CD-ROM interface	IDE or proprietary	SCSI
video	VGA	Super VGA
sound	16-bit SB-compatible	16-bit SB-compatible
power supply	150W	200W
DOS version	5	6.x

UNRAVELING THE MULTIMEDIA MYSTERY

If a simple upgrade isn't possible, and you don't want to buy a brand-new system, you can get do-it-yourself multimedia by choosing one of many upgrade kits on the market. Expect to pay as little as $240 and as much as $600, depending on audio quality and number of bundled software titles. When you purchase a kit, you'll get a CD-ROM drive with 16-bit sound capability, a sound card, speakers, and a microphone, plus in most cases several CD-ROM games, references, or education tools.

Most CD-ROM drives sold in kits run off a controller on the sound card; the drive attaches to it via a ribbon cable. Installation usually isn't difficult. But before you buy any kit, look inside your PC to make sure you have an open 16-bit ISA expansion slot and an available bay for the CD-ROM drive. If you don't have an open 16-bit slot, consolidate the functions of two or more cards by pur-

chasing a multifunction card, combining extra ports with, say, extra system-memory functions so that you can run your CD-ROM drive along with your system's existing 16-bit peripheral. Not having an available drive bay is a simpler problem to solve: Look for a kit offering an external CD-ROM drive, one that can attach to either a parallel (printer) port or to a SCSI port. (Note that parallel ports transfer data slowly.) You'll also need an external unit if your desktop PC stands on its side, tower-style. While most kits come with an internal CD-ROM drive, the Media Vision Premium Deluxe and the Plextor 4PLeX Sound Chaser are among those offering an external unit.

On the other hand, if you feel confident upgrading your system's components one by one, see the "Sound-Card Sampler" below for a selection of popular audio boards.

—M.N.

UPGRADE-KIT SAMPLER UNDER $500

Compro Multimedia Upgrade Kit, $340
ACS/KRIS Technologies, 800-282-5747

Digital Wave Express, $329
ALR, 714-581-6770

DSP-16 Sound System, $429
MediaMagic, 800-624-8654

Explorer Deluxe, $499
Sound Galaxy Voyager, $399
Aztech Labs Inc., 510-623-8988

Golden Sound Multimedia Kits, $350 and up
Toptek Technologies Inc., 800-416-8889

IDCDKIT1, $329
Identity Systems Technology, 800-723-8258

Megamedia SS20, $379
Megamedia Computer Corp., 800-634-2633

MV1100, 2100, 3100, 4100, $299–$599
Super Deluxe Multimedia Kit, $499
MediaVision Inc., 510-770-8600

Quadspeed ComPro Multimedia Upgrade Kit, $279
ACS Computer Group, 408-481-9988

Vienna, $375
Ocean Information Systems, 800-325-2496

UPGRADE-KIT SAMPLER $500–$1000

4PLeX Sound Chaser, $749
Plextor, 800-886-3935

Game Blaster CD 16, $549
Sound Blaster Digital Edge 3X, $999
Sound Blaster Edutainment CD 16, $599
Creative Labs Inc., 800-998-5227

MultiSound, $599
MultiSound Monterey, $899
Turtle Beach Systems, 717-767-0200

Premium Deluxe Multimedia Kit, $599
MediaVision Inc., 510-770-8600

Quad Speed Diamond Multimedia Kit 5000, $599
Diamond Multimedia Systems, 408-325-7000

Quantum, $599
Reveal Computer Products, 800-326-2222

RealMagic CD-ROM Upgrade Kit, $799
Sigma Designs Inc., 800-845-8086

Toptek Multimedia Advanced Upgrade Kit, $899
Toptek Technologies Inc., 800-416-8889

SOUND-CARD SAMPLER

Gallant SC 7000, $189
CompuMedia Technology, 510-656-9811

Golden Sound Pro 16 Plus, $299
Toptek Technologies Inc., 800-416-8889

MediaVision Pro 3D, $379
MediaVision Inc., 510-770-8600

Monte Carlo/Tropez Pro, $149 and up
Turtle Beach Systems, 717-767-0200

MSC Wave Pro 16, $299
Ad Lib Multimedia Inc., 418-656-8742

Sound Blaster 16 Value Edition, $139.95
Sound Blaster 16 MultiCD, $199.95
Sound Blaster 16 SCSI-2, $249.95
Sound Blaster AWE32, $399.95
Creative Labs Inc., 800-998-5227

Sound Galaxy Series, $159–$179
Aztech Labs Inc., 510-623-8988

TeleWave Audio 32, $299
TeleVideo Systems Inc., 408-954-8333

---

floppy disk. The boot disk contains your CONFIG.SYS and AUTOEXEC.BAT files, which load only the system software DOS needs to run that particular application. While the

CD-ROM's documentation typically provides a utility to create the boot disk, you must manually add the device-driver lines for your sound card and CD-ROM drive.

Hot DOS Multimedia Software
Once you've upgraded, sample some of the many great multimedia DOS applications out there, including a few of the best CD-ROMs:

Armored Fist, NovaLogic (818-880-1997): A sophisticated tank simulation with missions of varying difficulty and a realistic voice track.

Creature Shock, Virgin Interactive (714-833-8710): A Doom-like game that pits you against weird aliens aboard a derelict spaceship. Fantastic graphics and sound track.

Cyclemania, Accolade (408-296-8400): A motorcycle-racing game that pits you against other racers, farm animals, and oncoming traffic.

The Discoverers, Knowledge Adventure (818-542-4200): Teaches kids about famous explorers and scientists through videos and games.

Flash Traffic, Time-Warner Interactive (818-955-9999): Play an FBI agent working against the clock to stop terrorists from nuking Los Angeles.

Mad Dog McCree I and **II,** American Laser Games (505-880-1718): Just like the arcade. Match your shooting skills against some of the meanest hombres in the West.

Noctropolis, Electronic Arts (415-572-2787): Play a superhero who must rid the city of an evil force. Contains violence and some nudity.

The Picture Atlas of the World, National Geographic Society (800-368-2728): A multimedia atlas with up-to-date maps and videos.

Quarantine, GameTek (305-935-3995): Drive your heavily armed cab through a prison city. Run over crazies and ram rude drivers while listening to alternative rock music.

Return to Zork, Activision (800-477-3650): The adventure classic as an interactive movie. Few others can match its wit and challenge.

Under a Killing Moon, Access Software (800-800-4880): An interactive movie featuring an all-star cast, including Brian Keith and Margot Kidder. You play a bumbling P.I. in the 21st century out to save the world from an evil enemy.

None of the products listed here runs under Windows—tell that to the next Windows user who brags about his or her multimedia PC. ∎

KIT CAVEATS . . .

Just put in a sound card, add a CD-ROM drive, install your multimedia software, and go. If you believe that, you're in trouble.

Exciting and dazzling CD-ROM applications may be emerging daily, but multimedia is far from an exact science. In fact, installation software that comes with multimedia kits can be moody, freezing your system and sending you scrambling for an "all-our-representatives-are-busy-please-hold" customer-support number. Sure, taking the cover off your PC is easy. Plugging a sound card into a ready-and-waiting 16-bit slot is a piece of cake. Installing an internal CD-ROM drive into cramped quarters may pinch your wrist a bit, but nothing to sweat about. But just when you start to believe the "it's-so-easy" propaganda on your upgrade-kit box, you have to face the ultimate multimedia-kit task—loading the software that gets your PC multimedia-ready.

Popping in a 3.5-inch disk or two and typing INSTALL at the drive prompt sounds simple enough, but count yourself among the lucky if your installation goes without a hitch. If your PC already has a network card, a modem, or some other peripheral, be prepared for frustration. Installation software for some kits, for example, may recognize your new CD-ROM drive, but will lock up your system when it tries to activate your sound card. While the hang-up could be one of a dozen possibilities, chances are you're dealing with an IRQ (interrupt request) conflict. The kit's documentation should tell you how to rectify the problem, but it's not always that simple. Some kit sound cards come with no alternative IRQ; you have to fiddle around with the settings on your other peripherals to accommodate your stingy sound card.

Also, be ready to dive into CONFIG.SYS and AUTOEXEC.BAT (don't forget to make a backup) to jury-rig I/O ports and DMA settings, coordinate IRQs, and maybe even move your entire sound-card information to a different spot. Some kits, for example, may freeze your system if the automatic installation places the sound-card info after your network info, but will work fine if you go into CONFIG.SYS and move the sound-card line ahead of the network.

If it's not an IRQ or memory conflict, then the trouble might be the system itself. It was only after poring over the troubleshooting section of one kit's documentation that we learned that our PC—a widely known, commercial 486—probably wouldn't get along very well with the kit's installation. (Perhaps we should have seen trouble on the horizon when we first looked at the docs: two pages of installation instructions and some 40 pages of troubleshooting tips.) Fortunately, the manufacturer had a "fix" we could download from the company's customer-service BBS, but even that didn't get us up and running. Some ten phone calls and five representatives later, they threw up their hands and so did we. The kit is still in its box. That's not to say that disaster lurks for every multimedia-kit installation, but the message is clear: Before you buy any kit, check with your dealer or manufacturer to make sure it has no potential conflict with your system, and especially that its sound card offers more than one IRQ option.

If you think upgrading to a multimedia 486 will solve all your installation problems, think again. Ready-to-go multimedia PCs can be temperamental, too, but in their case, the issue is often with the application. You may find, as we have, that when you try to install a CD-ROM that's top-heavy with video, your system may cooperate up to a point and then freeze. Your screen may tell you what the problem is (and you thought you bought this thing ready to go!), but few CD-ROM products come with trouble-shooting booklets. You could return the "faulty" disc and get your money back, but that may not be the solution. In fact, you may have to live with the fact that your particular multimedia machine won't load certain CD-ROMs, while PC belonging to the guy down the street has little problem with the same product. But don't worry, he may give you some of his discs that won't work—until you get them running no-sweat on *your* PC!

Go figure. Well, maybe you have already, or maybe you've had a multimedia experience like ours. If so, tell us your advice or war story by faxing it to *DOS World*, 1-603-924-6972.

—Steve Smith, Senior Editor

MAKING MUSIC IN DOS

DOS's Merry Melodies

*Roll over, Beethoven! It may not be the Philharmonic,
but this full-featured music utility plays single-voice tunes
through your PC's speaker—in three different ways.*

by Robert L. Hummel

For years, the only sound my PC made came from the power-supply fan. Even today, unless you've plunked down a hundred-plus bucks for a sound card and speakers, your PC sits there, mute.

Still, even with a sound card, you're faced with learning to program the card directly—assuming that your sound card even came with the extensive documentation you'd need (unlikely) and that you're predisposed to self-torture.

Fortunately, there's an alternative. If you don't have expensive sound equipment or if you want to add a little music to your computing as simply as possible, you can use PLAYTUNE, a full-featured utility you can create with DOS's Debug. I introduced a scaled-down version of PLAYTUNE in the March 1995 "PC Toolbox" ("Play Music For Me," *DOS World* #20, page 32) as a utility for adding simple sounds and songs

Contributing Editor Robert L. Hummel is the author of PC Magazine Programmer's Technical Reference: The Processor and Coprocessor *and* PC Magazine Assembly Language Lab Notes *(Ziff-Davis Press).*

to batch files. It interprets ordinary text files and plays music through your PC's built-in speaker. This new version is versatile enough to accept input read from an external file, typed at the keyboard, or specified on the command line.

The PLAYTUNE Utility

The PLAYTUNE utility is based on Basic's PLAY statement, and uses a compatible superset of command arguments. Even so, the executable program PLAYTUNE.COM is less than 700 bytes. PLAY includes support for sharps, flats, tempo control, and dotted notes, and can play tones over seven octaves. The accompanying table (page 50, top) lists the commands PLAYTUNE accepts.

Before you can use PLAYTUNE, you'll have to create it. To convert the Debug script PLAYTUNE.SCR, shown in the first listing (opposite), to the executable program PLAYTUNE.COM, follow the instructions in the "Debug Scripts" section of "How to Use This Magazine" (page 71).

Interactive PLAYTUNE

You can use three methods to create music with PLAYTUNE. The simplest way is to enter notes from your PC's keyboard. Begin the program by typing PLAYTUNE at the DOS command prompt. From that point on, PLAYTUNE will try to interpret everything you type as a command. It ignores invalid music characters.

For example, execute PLAYTUNE and then type the letter A (uppercase or lowercase) and press the Enter key. You'll hear a beep from your speaker. The tone will be at 440Hz and last for a half-second. In the music world, 440Hz is the frequency that defines the pitch "concert A," the tone to which most instruments are tuned. It may be only a coincidence, but it's the same tone most PCs use for their default beeps.
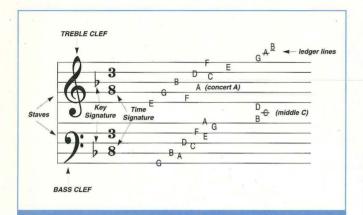
You can continue to type notes and other commands directly into PLAYTUNE. You'll need to press Enter at the end of each line before PLAYTUNE will interpret them. (This is a quirk of DOS, not PLAYTUNE.) Try changing the octave or the length of the notes using the O and L commands to see their effect. For an idea of what you can do, type the following line into PLAYTUNE, and press Enter:

PLAYTUNE.SCR creates the program PLAYTUNE.COM, which produces music and sound effects through your PC's speaker.

```
N PLAYTUNE.COM          CBW                     JZ   240                JMP   2A4               JZ    338
A 100                   MOV  BX,AX              MOV  [134],AL           MOV   AH,3F             MOV   [134],AL
JMP  138                MOV  AX,[126]           JMP  172                SUB   BX,BX             MOV   CL,[12D]
DW 0C0A 0301 0605       SUB  DX,DX              MOV  [12D],CL           MOV   CX,1              SUB   CH,CH
DW 0008 4B26 4746       DIV  BX                 JMP  172                MOV   DX,135            JCXZ  356
DW 4343 3F7C 3BEC       MOV  BX,AX              CMP  AL,54              INT   21               MOV   DX,BX
DW 388F 3563 3263       CALL 36D               JNZ  244                CMP   AX,CX             SHR   DX,CL
DW 2F8F 2CE4 2A5F       JMP  172                CALL 310                MOV   AL,[135]          SUB   BX,DX
DW 27FE 25C0 23A3       CMP  AL,41              CMP  AL,20              JMP   298              IN    AL,61
DW 1233 0004 048C       JB   1DA               JB   0244               MOV   BL,AL             OR    AL,3
DW 0203 0000 0000       CMP  AL,47              MOV  BX,AX              CALL  26A              OUT   61,AL
DW 0000 0000 FFFF       JA   1DA               MOV  AX,8880            JB    2F0              CALL  36D
MOV  AX,3523            SUB  AL,41              MOV  DX,8               CMP   AL,2D            IN    AL,61
INT  21                 MOV  BX,0102            DIV  BX                 JNZ   2E1              AND   AL,FC
MOV  [130],BX           XLAT                    MOV  [126],AX           DEC   BL               OUT   61,AL
MOV  [132],ES           CALL 2D2                SUB  DX,DX              JMP   2F0              JCXZ  36C
MOV  AX,2523            CALL 2F3                DIV  Word Ptr [128]     CMP   AL,2B            MOV   BX,DX
MOV  DX,39E             CALL 330                MOV  [12A],AX           JZ    2E9             CALL  36D
INT  21                 JMP  0172               JMP  244                CMP   AL,23            RET
MOV  AX,3508            CMP  AL,4C              PUSH BX                 JNZ   2ED             PUSH  CX
INT  21                 JNZ  1FC               PUSH CX                 INC   BL               MOV   CL,5
MOV  [395],BX           CALL 310                PUSH DX                 JMP   2F0              SHR   BX,CL
MOV  [397],ES           CMP  AL,40              CMP  Byte Ptr [137],FF  MOV   [134],AL         JNZ   375
MOV  AX,2508            JA   172                JNZ  28D                MOV   AL,BL             INC   BX
MOV  DX,383             OR   AL,AL              PUSH SI                 RET                    MOV   Word Ptr [12E],0
INT  21                 JZ   172               INC  Word Ptr [136]      SUB   BX,BX             CMP   BX,[12E]
MOV  AL,36              CBW                     MOV  SI,81              MOV   BL,AL             JA    37B
OUT  43,AL              MOV  [128],AX           LODSB                   ADD   BL,BL            POP   CX
JMP  168                MOV  BX,AX              CMP  AL,20              MOV   BX,[BX+10A]       RET
SUB  AL,AL              MOV  AX,[126]           JZ   27C                MOV   CL,[12C]         STI
OUT  40,AL              SUB  DX,DX              DEC  SI                 SHR   BX,CL           PUSH  AX
JMP  16E               DIV  BX                 MOV  BX,SI              MOV   AL,B6            CS:
MOV  AL,40              MOV  [12A],AX           POP  SI                 OUT   43,AL           INC   Word Ptr [12E]
OUT  40,AL              JMP  172                CMP  AL,D               MOV   AX,BX           CS:
CALL 26A                CMP  AL,4F              JZ   28D                OUT   42,AL           ADD   Byte Ptr [109],40
JNB  199                JNZ  205               MOV  [136],BX           MOV   AL,AH            JNZ   399
MOV  AX,2508            CALL 310                SUB  AL,AL              OUT   42,AL           POP   AX
LDS  DX,[395]           JMP  215                XCHG AL,[134]           RET                   CLI
INT  21                 CMP  AL,3E              OR   AL,AL              MOV   BX,A00          JMP   0:0
MOV  AL,36              JZ   20D                JZ   2A8                CALL  26A              MOV   AL,20
OUT  43,AL              CMP  AL,3C              CLC                     JB    32D             OUT   20,AL
JMP  186                JNZ  21F               PUSHF                   CMP   AL,30            POP   AX
SUB  AL,AL              SUB  AL,3D              CMP  AL,61              JB    32A             IRET
OUT  40,AL              ADD  AL,[12C]           JB   2A3                CMP   AL,39
JMP  18C                JS   21C                CMP  AL,7A              JA    32A             RCX
OUT  40,AL              CMP  AL,6               JA   2A3                SUB   AL,30           29F
MOV  AX,2523            JA   21C                AND  AL,DF              XCHG  AL,BL            W
CS:                    MOV  [12C],AL           POPF                    MUL   BH              Q
LDS  DX,[130]           JMP  172                POP  DX                 ADD   BL,AL                    End
INT  21                 CMP  AL,4D              POP  CX                 JMP   313
RET                     JNZ  247               POP  BX                 MOV   [134],AL
CMP  AL,20              CALL 26A                RET                     MOV   AL,BL
JZ   172                JB   244               MOV  BX,[136]           RET
CMP  AL,50              SUB  CL,CL              OR   BX,BX              MOV   BX,[12A]
JNZ  1C1                CMP  AL,4C              JZ   02BF               MOV   CX,BX
CALL 310                JZ   240               MOV  AL,[BX]            JMP   33C
CMP  AL,40              MOV  CL,3               CMP  AL,D               SHR   CX,1
JA   172                CMP  AL,4E              JZ   2BC                ADD   BX,CX
MOV  BX,[12A]           JZ   240               INC  Word Ptr [136]    CALL  26A
OR   AL,AL              DEC  CL                 JMP  297                JB    348
JZ   1BC                CMP  AL,53              STC                     CMP   AL,2E
```

A MUSICAL ROAD MAP

The music staff is made up of five horizontal lines and four spaces. Each line or space corresponds to one particular note. The first illustration (below) shows two staves connected by a line. The first symbol on each staff indicates the clef range—*treble clef* on the upper and *bass clef* on the lower in this example. The clef defines the range of notes assigned to each line or space as shown in the first illustration.

Grand staff: treble at top, bass at bottom.
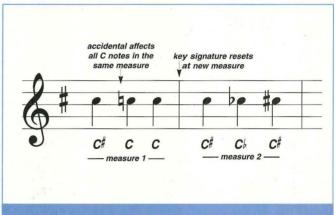
As you can see, the assignments for the treble and bass clefs are different, and remembering them can be difficult. For that reason, every music student learns that the notes on the treble-clef spaces spell *FACE* from bottom to top. On the bass clef, the spaces from bottom to top spell *ACEG*—easy to remember as *All Cows Eat Grass*.

You use *ledger lines* to write notes that would appear above or below a staff. The A and B notes that appear above the treble clef in the first illustration, for example, use ledger lines to indicate their positions relative to the staff. Also note that the treble and bass clefs are separated by one ledger line. The note that appears there is known as middle C.

The *key signature* appears immediately to the right of the clef on every line of music, and is simply a series of *accidentals* (sharps, flats, and naturals) on specific lines or spaces of the staff. When a sharp or flat is used in a key signature, all occurrences of the note represented by that line or space are automatically sharped or flatted. In the first illustration, the key signature shows a flat on the B note (on both treble and bass clefs). Thus, a B note appearing anywhere in the music should be interpreted as a B-flat (even if it appears above or below the staff on a ledger line.)

Although the key signature establishes a default key, you can still override the default for individual notes. An individual note

Accidentals (sharps, flats, and naturals) change the pitch of notes.

You'll be a darling when you play CLEMEN-.NOT. Note that the listing contains the letter O (not zero) in several places.

```
O3 L8 C. L16 C
L4 C O2 G L8 O3 E. L16 E
L4 E C L8 C. L16 E
L4 G. L8 G F E
L2 D L8 D. L16 E
L4 F F L8 E. L16 D
L4 E C L8 C. L16 E
L4 D O2 G L8 B O3 D L2 C.

O3 L8 C. L16 C
L4 C O2 G L8 O3 E. L16 E
L4 E C L8 C. L16 E
L4 G. L8 G F E
L2 D L8 D. L16 E
L4 F F L8 E. L16 D
L4 E C L8 C. L16 E
L4 D O2 G L8 B O3 D L2 C
```
End

This file, TURKEY.NOT, plays a lively rendition of *Turkey in the Straw.*
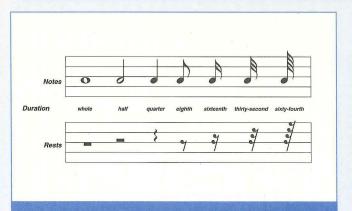
```
T255
MN L8 B A
MS L4 G G MN L8 G E O2 B O3 C
D E D C# L4 D MN L8 G A
MS L4 B B MN L8 B A G A
MS L4 B A A MN L8 B A
MS L4 G G MN L8 G E O2 B O3 C
D E D C# L4 D MN L8 G A
MS L4 B O4 D MS L8 D O3 B G A
MS L4 B A ML L2 G
MN L4 O2 B MS O3 D MN O2 B MS O3 D
MN L8 O2 B O3 C MS L4 D ML L2 D
MN L4 C MS E MN C MS E
MN L8 C O3 D MS L4 E E MN D
MS G G D D O2 B B A O3 L8 G A
MS L4 B O4 D MS L8 D O3 B G A
MS L4 B A G
```
End

This version of *Reveille*, REVEILLE.NOT, will have you saluting in the aisles. Note the single zero (slashed) in the first line. Also, lines 2 and 3 show overflow.

```
T240
L4 CF L8 AF L4 CAF L8 AF L4 CAF
    L8 AF L4 CF L2 A L4 F
L4 CF L8 AF L4 CAF L8 AF L4 CAF
    L8 AF L4 CC L2 F.
L4 AAAAA O4 L2 C O3 L4 A
 FAFAF  L2 A L4 F
 AAAAA O4 L2 C O3 L4 A
L8 FA L4 O4 C O3 CCC L2 F
```
End

Symbols for notes and rests you're likely to run across when reading music.

symbols that are typical of those in most music. Eighth (and shorter) notes are often shown joined together, as in the fourth illustration (below). In the first example, two eighth notes are joined by a bar for notational convenience, but are played as separate notes. In the second example, a dotted eighth note is joined to a sixteenth note. The dot following a note or rest increases its duration by half its previous value. In this example, the eighth note followed by a dot has a value of three-sixteenths: one-eighth plus one-sixteenth.

In the final example, a half note is joined to a quarter note with a curved line called a *ligature*. This indicates that the two notes should be played as a single phrase. In this case, because the notes are the same, the effect is the same as a single three-quarter note (a dotted half note). A ligature is often necessary when a single note has to span a measure.

—R.L.H.

may be prefaced with a flat, sharp, or natural. The accidental affects all subsequent notes of that type in the measure in which it appears. The default key signature automatically resumes when the measure ends. The second illustration (facing page, right) shows an example of how accidentals are used.

The *time signature*, shown in the first illustration, indicates rhythm and meter. The upper numeral indicates the number of beats in a measure. The lower numeral indicates which type of note receives one beat. In this example, the 8 means that an eighth note gets one beat. The 3 means that the notes and rests in each measure (a section of the staff between two vertical lines) will add up to three beats. Other common time signatures are 4/4 (called *common time* and often shown as simply *C*), 2/2, and 6/8. Rhythm and meter are simply organizational conventions and are independent of tempo.

Specific symbols indicate duration of notes and rests. Although there are minor variations, the third illustration (above) shows

Straight lines join separate notes for appearance, but ligatures create a single note.

```
03 L16 CP CC DP CP PE PF
```

When you're done, terminate the program by pressing Ctrl+Z, Enter.

Single-Line PLAYTUNE

Using PLAYTUNE interactively is great for trying out a tune or sound effect. But typing in your masterpiece every time you want to hear it gets old fast. Fortunately, PLAYTUNE supports a single-line command mode that lets you embed sounds in your batch files. Here's the syntax:

```
PLAYTUNE string
```

where *string* is any combination of PLAYTUNE commands—with two important exceptions. In single-line mode, never use the octave-change characters < and >. Before PLAYTUNE gets control, DOS sees these characters and attempts to interpret them as redirection characters. Consider this example:

```
PLAYTUNE CEG>A
```

Instead of playing four notes, this command line will play the first three notes and create a zero-length file named A.

Single-line mode is handy for adding sound effects to your batch files. For example, the following line produces the attention-getting sound of an alien plasma rifle:

```
PLAYTUNE L64CDEFGABCDEFGABCDEFGAB
```

Redirected PLAYTUNE

Sure, running PLAYTUNE at the command line is an interesting way to while away the hours. And single-line PLAYTUNE is great for short

```
Twilight Zone.
L8 O4 FF+FD FF+FD FF+FD

Traditional phone.
L32 CF CF CF CF CF CF CF CF L2P
L32 CF CF CF CF CF CF CF CF L2P

Upward trill.
L16 C C+ D D+ E F F+ G G+ A A+ B

Downward trill.
L16 B A+ A G+ G F+ F E D+ D C+ C
```

COMMANDS FOR THE PLAYTUNE UTILITY

A-G	Plays a note in the range A through G. Notes may be modified with #, +, and - symbols.
+,#	Follows a note, changing it to a sharp.
-	Follows a note, changing it to a flat.
.	Follows a note or rest, increasing the duration by half the previous value.
>	Switches to the next-higher octave. (Don't use this symbol when typing command strings at the DOS prompt; DOS will interpret it as redirection.)
<	Switches to the next-lower octave. (Don't use this symbol when typing command strings at the DOS prompt; DOS will interpret it as redirection.)
P[n]	Indicates musical rest (pause). No sound is produced. Range for n may be set for 1 to 64. If n isn't specified, duration of rest is length set by most recent L command.
Ln	Sets length of each note or pause. L1 is a whole note. L4 is a quarter note. Range for n is 1 to 64. After length

	is set, it applies to all subsequent notes and rests until another L command is encountered.
On	Sets octave. Range for n is 0 (zero) to 6. Default octave is O3, which contains middle C (256Hz) and concert A (440Hz). An octave affects all subsequent notes until another O command is encountered.
Tn	Sets tempo, measured in quarter notes per minute. Range for n is 32 to 255. Default tempo is 120 quarter notes per minute (one whole note = 2 seconds).
MN	Music Normal. Plays each note for seven-eighths of the note's duration, followed by a period of silence lasting one-eighth of the note's duration.
MS	Music Staccato. Plays each note for three-quarters of the note's duration, followed by a period of silence lasting one-quarter of the note's duration.
ML	Music Legato. Plays each note for the full length of the note, with no pause between notes.

sounds. But PLAYTUNE shines when you use it with a command file.

Because PLAYTUNE reads from standard DOS input, you can tell it to read its input from a file instead of the keyboard, using the DOS redirection command. For example, if you yearn for folk songs, create the CLEMEN.NOT file shown in the second listing (page 48, far left). Then use the following command to play it:
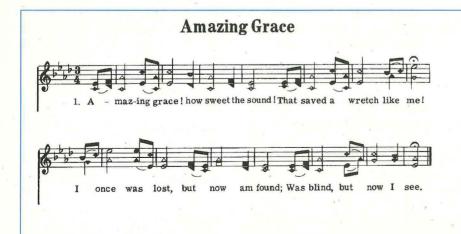
```
PLAYTUNE < CLEMEN.NOT
```

On Your Own

I'm no composer, but I've put together a few command files that play songs you should recognize. TURKEY.NOT, shown in the third listing (page 48,

middle), plays my personal rendition of *Turkey in the Straw*. Or you can wake yourself up with the REVEILLE.NOT file, shown in the fourth listing (page 48, right). And the fifth listing, actually a collection of short listings (page 49), shows a few sound effects you can use as starting points for your own creations. I've added spaces and carriage returns in the NOT files for clarity only—PLAYTUNE doesn't interpret them as commands. If you pack the commands, you can get up to 123 characters on a single line.

After you've created PLAYTUNE.COM and played the sample tunes, you'll probably want to create some music files of your own. Printed music can

be a great source of entertainment; if you're feeling rusty, the accompanying sidebar, "A Musical Road Map" (page 48), describes the fundamentals of musical notation. Remember that sheet music is simply a shorthand way of representing specific notes and their durations. As for translating sheet music into PLAYTUNE format, the accompanying illustration (below) shows the score for the hymn *Amazing Grace*, along with the corresponding PLAYTUNE commands for the melody line, measure by measure. Note that there are eight measures in the first line and eight in the second, numbered 1 through 16 in the table. Rise up and sing! ■

Amazing Grace

1. A - maz-ing grace! how sweet the sound! That saved a wretch like me!

I once was lost, but now am found; Was blind, but now I see.

```
 1. T90 O3 L8 ML E- MN F
 2. L2 A- L8 ML O4 C MN O3 A-
 3. L2 O4 C L4 O3 B-
 4. L2 A- L4 F
 5. L2 E- L8 ML E- MN F
 6. L2 A- L8 ML O4 C MN O3 A-
 7. L2 O4 C L8 ML O3 B- MN O4 C
 8. L2 E-.
 9. L8 ML O3 B- MN O4 C
10. L2 E- L8 ML E- MN C
11. L2 O3 A- L8 ML F MN E-
12. L2 A- L4 F
13. L2 E- L8 ML E- MN F
14. L2 A- L8 ML O4 C MN O3 A-
15. L2 O4 C L4 O3 B-
16. L2 A-.
```

Caching In On Performance

A disk cache is an easy, effective way to improve your PC's performance. Two popular tools, Super PC-Kwik and MS-DOS's built-in SmartDrive, can make a slow hard disk seem fast and a fast hard disk seem even speedier.

by Kenneth E. Johnson

The time your computer spends accessing your hard-disk drive drags down the system's overall performance—but a technique called *caching* can pick up the pace.

A disk cache works by creating an area of RAM in which to store disk data temporarily. Then when DOS requests information from the disk, caching software reads whole tracks of disk data at a time, storing it in this stash.

Chances are that additional requests will require data already in the cache, which DOS can then retrieve quickly from RAM, instead of accessing the disk again.

Most disk caches also do write caching; when DOS writes data to the disk, the cache program intercepts it and stores it temporarily. The cache then waits until system resources aren't busy, or a few seconds pass, before writing the data.

By delaying writes, the cache gives priority to reading data and to other processing, while writes wait until there's a lull in your work.

Contributing Editor Kenneth E. Johnson is training and support manager at the law firm of Mayer, Brown & Platt in Chicago. He's a columnist and writer for several business and computer-law publications and serves as a contributing editor for the American Bar Association's NETWORK 2d *newsletter.*

Two popular disk caches are SmartDrive, which comes with MS-DOS and Microsoft Windows, and Super PC-Kwik 6.04, from PC-Kwik Corporation ($79.95; 800-274-5945, 503-644-5644). Which one might meet your particular computing needs? Let's see how you can benefit from these two programs— while avoiding any unpleasant surprises.

SMART COMPUTING

SmartDrive (SMARTDRV.EXE) is the disk cache you probably already have. A version of SmartDrive has been included with MS-DOS since version 5, and with Microsoft Windows since version 3.0.

The Windows and MS-DOS upgrades in the last few years have automatically installed SmartDrive for you—a clear indication of the importance of disk caching to system performance.

TIP #1 — Be Respectful of Write Caching

With write caching, if the computer freezes or is shut off before the data in the cache is written back to the disk, it will be irretrievably lost. It's also possible that if the cache is updating the disk's system areas when the computer shuts down, other data on the disk can

be corrupted. If that happens while your computer is writing to a DoubleSpace or DriveSpace compressed disk, all data on that disk could become inaccessible.

This is a serious shortcoming, because by default SmartDrive write-caches all hard disks and so can put your data at risk without your knowledge. SmartDrive 4.0 (which came with Windows 3.1) and 4.1 (which shipped with MS-DOS 6.0) not only have write caching enabled, but also immediately return control to DOS after caching. That means that you can see the DOS prompt and think it's safe to turn off your computer, even though SmartDrive hasn't flushed (that is, written) the cache yet.

An uproar erupted over the default caching of MS-DOS 6.0's SmartDrive 4.1, and in June 1993 Microsoft released SmartDrive 4.2, which included a switch to turn off write caching for all drives. In addition, it flushed the cache before returning the DOS prompt, so there was less chance that you'd turn off the PC while data was still waiting to be written.

MS-DOS 6.2 brought us SmartDrive 5.0, which similarly can disable all write caching and also won't return the DOS prompt before the cache is written to disk.

That said, all versions of SMARTDRV.EXE let you turn off write caching by including the drive letters (without colons) on the SmartDrive command line in your AUTOEXEC.BAT file.

For example, SMARTDRV C enables read caching and disables write caching for drive C. If you're using SmartDrive 4.2 or 5.0 and want to disable write caching for all drives, include the /x switch in the command as follows: SMARTDRV /X.

If you try to install MS-DOS 6.0's DoubleSpace or MS-DOS 6.22's DriveSpace on a computer with SmartDrive running, you may receive the error message "You must specify the host drive for a Double-Space drive." This occurs when write caching is enabled for the drive you're trying to compress. Turn off write caching for the drive, then run the Double-Space/DriveSpace installation again.

SmartDrive doesn't cache compressed drives, but rather the host drive containing the compressed volume file (for example, DBLSPACE.000). Caching the host also caches the compressed drive. But because your system may lose and corrupt data if it goes down before the cache is written, most experts recommend disabling write caching of host drives.

TIP #2
Flush That Cache

If you do use write caching, make sure the cache is flushed before turning off your PC. This will ensure that no data is lost. Any number of circumstances may make SmartDrive write-cache data immediately to the disk. You can flush the cache with the command SMARTDRV /C.

If you use batch files to run your applications, add that command at the end of each batch file to force SmartDrive to flush any pending writes before you get the DOS prompt back.

It's important to include SMART-DRV /C in any batch files that reboot your PC, because there's no significant pause between execution of commands in a batch file; Smart-Drive may not have time to write the cache data before DOS executes the reboot command. Tell Smart-Drive to flush the cache immediately, before any reboot command.

SmartDrive also writes the cache information if you use Ctrl+Alt+Del to do a warm boot. SmartDrive intercepts DOS's famous "three-fingered salute" and flushes the cache before it lets the computer reboot.

Unfortunately, it can't flush the cache if you simply turn off the power or press your computer's

BEATING THE EMPTY-DRIVE BLUES

Ever since the introduction of DOS 2.0, users have put PG into their PROMPT commands so that DOS would include the names of the current drive and directory ($P) and a greater-than sign ($G) in its prompt. As helpful as that is when navigating a complex directory structure, it can cause severe aggravation when you switch to an empty floppy. To display the $P portion of the prompt, DOS must read the disk to retrieve the name of the current directory.

To see why this is such an insidious problem, imagine that you switch to drive A and the drive is empty. DOS then issues the following error message:

```
Not ready reading drive A
Abort, Retry, Fail?
```

If you press A (for *abort*), you'll cancel the attempt to read the empty drive. DOS is ready for its next command, so it displays its prompt—or tries to. Because the prompt requires the current path, DOS reads the disk again and displays the error message again. So, you select *abort* again. DOS tries its prompt again, and so on.

The proper way to deal with this situation is to press F (for *fail*). DOS displays the following prompt:

```
Current drive is no longer valid>
```

You then may type C: to move back to a valid drive.

—Robert L. Hummel

```
C:\-->smartdrv
Microsoft SMARTDrive Disk Cache version 5.0
Copyright 1991,1993 Microsoft Corp.

Cache size:  2,097,152 bytes
Cache size while running Windows:  2,097,152 bytes

              Disk Caching Status
drive   read cache   write cache   buffering
---------------------------------------------
 A:        yes          no           no
 B:        yes          no           no
 C:        yes          no           no
 E:        yes          no           no
 F:        yes          no           no
Write behind data will be committed before command prompt return:

For help, type "Smartdrv /?".

C:\-->
```

DOS's SMARTDRV command screen, showing drives' cache size and caching status.

Reset button. (By the way, if you use Windows, Smart-Drive recognizes when you exit Windows and flushes the disk cache before returning to the DOS prompt.)

Perhaps the best way to ensure that the cache is flushed is to create a BYE.BAT file. It should look something like this:

```
@ECHO OFF
SMARTDRV /C
ECHO You can now safely shut off your PC
```

Now simply type BYE each time you're finished using your computer, and you'll be sure to preserve your data.

TIP #3 — Always Load VSAFE Before SmartDrive

Normally, SmartDrive writes its cache data to disk whenever you press Ctrl+Alt+Del. Unfortunately, it *doesn't* flush the cache if VSAFE (Microsoft's antivirus TSR program) loads after SMARTDRV. As a result, you can lose data, because VSAFE intercepts the Ctrl+Alt+Del call and doesn't pass it on to SmartDrive. The problem won't occur if VSAFE loads before SMARTDRV, so make sure you place the VSAFE command line before SMARTDRV in your AUTOEXEC.BAT file.

TIP #4 — Reduce Buffers In CONFIG.SYS

DOS uses its own read/write buffers to hold data when reading from and writing to the disk. Before accessing the disk, DOS checks its buffers to see whether the data needed is already there. DOS's buffers are like a small cache, but they're not nearly as efficient as a disk cache. Each buffer takes up about 530 bytes of RAM—not much room for data.

Unfortunately, DOS looks through all its buffers before going to the disk cache—but you can speed up performance by reducing the number of buffers it has

to search. With SmartDrive, about 15 DOS buffers are sufficient. Buffers are specified in CONFIG.SYS, so change the setting to BUFFERS=15.

TIP #5 — Use the Latest Version

Because various DOS and Windows upgrades have included SmartDrive, you should always use the most recent version of SMARTDRV.EXE on your system. Compare the dates of the various SmartDrive files in your \DOS and \WINDOWS subdirectories, and include the path for the latest version in the command in the AUTOEXEC.BAT file.

Unfortunately, SmartDrive may actually report the wrong version number when you use its functions from the command line. Instead of displaying the version loaded into memory, it will report the version of the first copy of SMARTDRV.EXE it finds on the path. To prevent that, rename or delete all older versions of SMARTDRV.EXE on your hard disk. If you've upgraded to MS-DOS 6.x, don't use SMARTDRV.SYS—it's Smart-Drive's version 4.0, which came with MS-DOS 5 and was loaded as a device driver in CONFIG.SYS.

Starting with version 4.1, which shipped with MS-DOS 6.0, SmartDrive is now the executable file SMART-DRV.EXE and is loaded in AUTOEXEC.BAT. Although you don't need it anymore, MS-DOS 6.x's Setup programs don't remove SMARTDRV.SYS from the \DOS directory.

SMARTDRV.SYS isn't compatible with MS-DOS 6.0 or 6.2 (you'll get an "Incorrect DOS Version" error), or with DoubleSpace or DriveSpace. Because SMARTDRV.SYS was replaced with SMARTDRV.EXE, you may delete it from your \DOS directory and use only SMARTDRV.EXE.

TIP #6 — Check the Cache Status

There are several ways to monitor the efficiency of the SmartDrive cache. Typing in SMARTDRV at the command
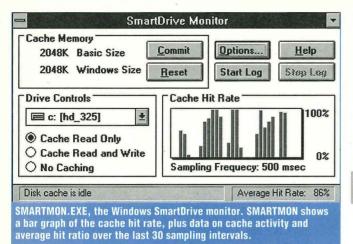
```
C:\-->smartdrv /s
Microsoft SMARTDrive Disk Cache version 5.0
Copyright 1991,1993 Microsoft Corp.

Room for    256 elements of   8,192 bytes each
There have been    6,297 cache hits
        and    1,577 cache misses

Cache size:  2,097,152 bytes
Cache size while running Windows:  2,097,152 bytes

              Disk Caching Status
drive   read cache   write cache   buffering
---------------------------------------------
 A:        yes          no           no
 B:        yes          no           no
 C:        yes          no           no
 E:        yes          no           no
 F:        yes          no           no
Write behind data will be committed before command prompt returns.

For help, type "Smartdrv /?".

C:\-->
```

SMARTDRV /S command screen, showing cache status and hits and misses. You calculate the "hit rate" by dividing number of hits by total hits plus misses, then multiplying by 100.

SMARTMON.EXE, the Windows SmartDrive monitor. SMARTMON shows a bar graph of the cache hit rate, plus data on cache activity and average hit ratio over the last 30 sampling intervals.

line displays information on the cache size and disk-caching status. (See the first screen shot, page 53, top left. Note that if SmartDrive isn't already running, this step loads it.) You'll see the normal cache size, the cache size when Windows is running, the read- and write-cache status for each drive, and whether double buffering is being performed on the drives. Double buffering provides cache compatibility for some hardware controllers, and may be needed for SCSI, ESDI, and MCA machines.

Entering SMARTDRV /S on the command line displays additional information about SmartDrive's status, including cache hits and misses. (See the second screen shot, page 53, bottom.) You can find the current cache "hit rate" by dividing the number of hits by the sum of hits and misses, then multiplying by 100. For example, if SMARTDRV /S shows 967 cache hits and 200 cache misses, the calculation would be (967/(967+200))*100: a hit rate of 82 percent.

You'll want to try for a hit rate of 60 to 80 percent. Less than 60 percent means that the cache isn't as effective as it could be, so you should make it larger. Much more than 80 percent typically means that more RAM is allocated to SmartDrive than necessary. You can make better use of your RAM by making it available to your programs.

If you're a Windows user, you can get cache information with SMARTMON.EXE, the Windows SmartDrive

Monitor included with MS-DOS 6.x. (See the third screen shot, left.) When SMARTMON runs minimized, it displays the cache-hit percentage. When maximized, it shows a bar graph with the cache-hit percentages of recent disk activity. The bottom of the SMARTMON window shows a status line with the activity of the cache and the average hit rate over the last 30 sampling intervals.

TIP #7 Watch for Syntax Errors

If a syntax error occurs when DOS tries to load SMARTDRV.EXE, SmartDrive won't load. Instead of giving an error message, DOS displays the SmartDrive on-line help screen, the same one you get by running SMARTDRV with the /? switch. Unfortunately, if this occurs while AUTOEXEC.BAT is executing, you may miss it as other programs load and think that SmartDrive is running when it's not.

Besides syntax errors, two switches will prevent SmartDrive from loading if you put them on the command line when starting SMARTDRV.EXE: /C and /S. The /C switch writes the cache to disk; the /S switch displays status information about the cache. Use them only when SMARTDRV is running.

QUICK OFF THE MARK

Super PC-Kwik is one of the premier third-party disk caches on the market. It has consistently been rated among the top disk caches for several years. It outperforms SmartDrive in all benchmark tests for read caching, write caching, and CD-ROM caching. Super PC-Kwik is available as a stand-alone product or as part of the PC-Kwik Power Packs for DOS and Windows ($99.95 each), and the WinMaster Windows utility kit ($129.95). Beginning with Super PC-Kwik 6.0, introduced in 1994, the program added a number of useful new features:

● By using DOS Protected Mode Services (DPMS), Super PC-Kwik can reduce its conventional-memory footprint to around 20K. If more than 22K of upper memory is available, Super PC-Kwik loads itself high and uses no conventional memory.
● Version 6.x offers faster CD-ROM caching than previous versions, also outperforming SmartDrive 5.0.
● The software also includes a Kwik-Load option to boot Windows faster.
● You can install Super PC-Kwik 6.x from DOS or Windows, along with a Windows configuration utility.

Now let's talk about how you can soup up your computing.

DOS in Windows TIP

FORMATTING DOUBLESPACE/ DRIVESPACE SYSTEM DISKS

When you use File Manager's Make System Disk option to format a floppy disk, Windows doesn't copy the DBLSPACE.BIN file you need to activate DoubleSpace or the DRVSPACE.BIN file you need to activate DriveSpace. If you want to create bootable floppy disks that will let you access compressed drives, get yourself out to the DOS prompt, then type FORMAT A: /S.

—Doug Lowe

Configure Super PC-Kwik In Windows

If you have Microsoft Windows, the easiest way to set up Super PC-Kwik is through its Windows configuration utility, CONFIG.EXE (located in the \PCKWIK subdirectory). This program makes changes to PCKWIK.INI, which Super PC-Kwik uses to set its parameters when loading. The Windows configuration program lets you change options by clicking on buttons and making selections from lists.

To set up Super PC-Kwik without Windows, you'll need to edit the PCKWIK.INI file in the \PCKWIK subdirectory. To change a setting, add the appropriate parameter and setting under the [SUPERPCK] section heading. (See the Super PC-Kwik manual for specific parameters.)

Super PC-Kwik's Measurements screen, displayed when you type SUPER /MEASUREMENTS on the command line.

Take Advantage of Write Caching If You Want It

Super PC-Kwik, unlike SmartDrive, purposely *doesn't* write-cache drives by default. You can, if you choose, have Super PC-Kwik do write caching. It makes Super PC-Kwik faster, at the risk of corrupting data if there's a system lockup or power failure.

If you're using the Super PC-Kwik Windows configuration program, at the Disk Drives window choose Advanced PowerWrite for the drive and set the Write Delay option. (PC-Kwik doesn't recommend going over two seconds.)

If you're editing the PCKWIK.INI file manually, add these two lines under the [SUPERPCK] heading:

```
PowerWrite=yes
WriteDelay=2.0
```

With write caching, make sure you flush the disk cache before turning off your PC. Super PC-Kwik provides two commands for that function. Entering SUPER /FLUSH at the command line writes the cache data to disk, clears the read-cache buffers, and resets the cache measurements to zero.

Alternatively, you may type SUPER /FORCEWRITES, which writes the waiting cache data but doesn't clear the buffers.

Install DPMS

As noted above, Super PC-Kwik 6.x uses DOS Protected Mode Services (DPMS) to reduce the amount of conventional memory it needs. DPMS, an interface provided by Novell Corporation, lets DOS load program code into extended (XMS) memory.

When you install Super PC-Kwik, you can choose to install DPMS. The installation program searches your system to see whether a DPMS server is already installed; if not, it copies one to your hard drive and adds the driver to CONFIG.SYS.

The DPMS driver takes up less than 2K of memory and can't be loaded high. If your memory manager's optimization program adds the DEVICEHIGH statement (or an equivalent command) to the DPMS.EXE line in CONFIG.SYS, you must remove it.

If your system locks up when DPMS loads, edit CONFIG.SYS and move the line containing the DPMS.EXE statement immediately after loading your memory manager (such as EMM386.EXE). DPMS works best when loaded directly after the memory manager.

Although DPMS may save you some conventional memory, it can also slow Super PC-Kwik. With sufficient conventional memory or enough upper memory to load all of Super PC-Kwik high, you'll probably want to run it without DPMS. In addition, Super PC-Kwik's DPMS conflicts with some application programs, including Borland C++, Quattro Pro, Paradox, Central Point Backup for Windows, and OzCIS. In such cases, remove the DEVICE=C:\PCKWIK\DPMS.EXE line from your CONFIG.SYS file.

Cache Your CD-ROMS

To cache a CD-ROM drive, you must load Super PC-Kwik after MSCDEX in AUTOEXEC.BAT. In addition, you must load PC-Kwik's PCKCDROM device driver after your CD-ROM device driver in CONFIG.SYS.

If PCKCDROM gives an error message indicating that it can't find your CD-ROM device, you may need to tell it the device name—it's located on the MSCDEX command line. For example, if the line reads C:\DOS \MSCDEX /D:MSCD001, add the line DEVICE=MSCD001 to the [CD-ROM] section of PCKWIK.INI.

A read-ahead buffer for the CD-ROM cache will help speed up data retrieval from the CD-ROM. Set it by adding BufferSize=nn to the [CD-ROM] section of the PCKWIK.INI file.

The valid values (nn) of the buffer are 0 (zero) to 48; the default is 8K. Beyond 24K, you find that it's a case

```
Super PC-Kwik(R) Disk Cache, Version 6.01 S/N 5051922500R10
Copyright 1986 - 1994 PC-Kwik Corporation, All Rights Reserved.
Licensed to: Kenneth E. Johnson

Super PC-Kwik is loaded and is now enabled.
Subset of program (SUPER.SB7) loaded.

Following is a list of the parameters in effect:

  /FDSupport        Standard support of diskette transfers.
  /HDSupport        Advanced read/write support for hard disks.
  /BatchRequests    Perform batch copies to/from cache.
  /ReadWritePriority Optional algorithm for advanced support.
  /QuickPrompt      Return DOS prompt normally.
  /TrackReads       Track buffering using a 17-sector buffer.
  /VolumeDetection  Use volume change hardware.
  /UniqueWrites     Do not check write requests for redundancy.
  /BusMasterDMA     Normal track buffer placement.
  /IgnoreDrive      Drive D cannot be cached -- physical unit unknown.
  /LoadHigh         Cache is loaded in high memory.
  /LendAmount       3,984K reserved for lending.
                         OK on loan.
                    3,984K available.
Press any key to continue...
```

Super PC-Kwik's Parameters screen, showing parameters currently in effect. You can display this screen by typing SUPER /PARAMETERS on the command line.

of diminishing returns; a larger buffer size won't really improve CD-ROM drive performance significantly.

TIP #5 Check Super PC-Kwik's Status

There are several ways to get information about the Super PC-Kwik cache, both at the DOS command line and in Windows.

If you're using DOS, typing SUPER /MEASUREMENTS displays cache statistics: logical transfers, physical transfers, transfers saved, and the percentage saved (hit rate). (See the fourth screen shot, page 55, top.) Typing SUPER /PARAMETERS displays all cache settings currently active. (See the fifth screen shot, above.)

You can also monitor cache information by including the following two lines in PCKWIK.INI, under the [SUPERPCK] heading:

```
HitIndicator=on
Popup=SHIFT
```

Hit Indicator flashes an asterisk character in the top-right corner of the screen whenever Super PC-Kwik gets information from the cache instead of the drive. Popup will display cache measurements when you press both Shift keys.

You can change this hot key to Alt+Shift by entering Popup=Alt, or to Alt+Ctrl +Shift by entering Popup =Ctrl. Both Popup and Hit Indicator work only at the DOS prompt or when you're running in text mode.

If you're using Windows, you can monitor the cache with SUPERWIN.EXE, Super PC-Kwik Meter. (See the sixth screen shot, right.) This program shows current cache size, location, and hit rate.

Clear, readable graphs show what disk activity has occurred in the past 40 seconds: The Logical Transfers chart shows programs' disk requests; Physical Transfers shows the actual number of requests that required disk access; Percentage Transfers from Cache illustrates the hit rate.

TIP #6 Disable Super PC-Kwik When Upgrading DOS

As a rule, it's a good idea to disable third-party disk caches when upgrading your DOS version. This prevents the possibility that the DOS Setup program will miss the existing disk cache and install SmartDrive, which will result in your running two caches simultaneously. Using multiple disk caches is a recipe for corrupting the file-allocation table (FAT) and damaging files.
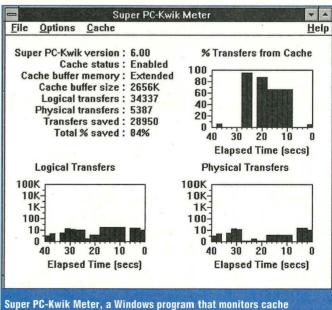
For example, MS-DOS 6.0's Setup didn't analyze AUTOEXEC.BAT lines beginning with the at sign (@). Earlier versions of PC-Kwik's Power Pack products used this line to load Super PC-Kwik:

```
@CALL C:\PCKWIK\SUPERON
```

DOS 6.0's Setup didn't detect the Super PC-Kwik cache and, therefore, also installed SmartDrive. Luckily, that doesn't occur with the MS-DOS 6.2 upgrade, because its Setup correctly analyzes lines starting with @ (although it does remove the @). For safety's sake, remark out (REM) the Super PC-Kwik command line in AUTOEXEC.BAT and reboot before upgrading your DOS version. After the upgrade, delete the SMARTDRV line added by the upgrade and "unremark" the Super PC-Kwik line.

REV IT UP

Disk caches can improve your work flow significantly, but as you can see, they can be a little tricky. Whether you use the "free" SmartDrive program, or Super PC-Kwik for its additional speed and features, you can easily "cache" in on improved speed and performance. Just approach write caching with a healthy dose of respect—and forewarning. ■

Super PC-Kwik Meter, a Windows program that monitors cache performance. "Total % saved" is the hit rate.

BACKGROUND EFFECTS

Your Attention, Please!

Brighten your programs by painting your screens with eye-catching wallpapers. QBasic's tile strings make it easy to design attractive displays.

by Greg Ennen

W hen your thoughts turn to redecoration, one of the first options you probably consider is wallpaper. The right choice of motif and color can transform a room, making it seem livelier and more inviting.

Carefully chosen computerized "wallpaper" can work similar magic with a QBasic program. Adding an interesting background to an opening screen welcomes users; combining several contrasting patterns can produce three-dimensional effects that capture and hold the eye.

Intrigued by the possibilities, I set out to explore the limits of creating wallpaper effects in QBasic. My success wasn't exactly immediate. Although my programming reference described how to use the PAINT statement to fill an object with a repeating pattern block, or *tile*, implementing this technique was tedious. Drawing a square, rectangle, or circle to hold the pattern was the easy part; creating a

Greg Ennen is a computer technician at a major daily newspaper in South Florida and also works as a free-lance writer for a number of computer publications.

satisfactory tile took time. Often, after working out a pixel-by-pixel map of my pattern block, converting it to a tile string, and painting the string into an object, I got a wallpaper whose design bore scant resemblance to the effect I'd envisioned originally.

Despite these setbacks, however, I remained fascinated with tiling, so I turned my attention to writing a tile-string generator. The final product, TILE.BAS (see the listing on page 59), not only eliminates surprises, it also handles the onerous task of converting a design to a tile string QBasic can interpret.

You fill in a design grid pixel by pixel, using any of the 16 colors available from QBasic's screen 9, then press a key to view the wallpaper effect your design will produce. If you like what you see, save the tile string in a file. (See the first screen shot, "All in one," page 58, top.)

From there, it's a simple matter of copying the file into a program and painting it into an object. My demonstration program, TILE-DEMO.BAS (page 62), shows how you might incorporate several tile strings created with TILE.BAS into a

single screen. (See the second screen shot, "Filling objects," page 58, bottom.)

Program Creation

You may download both listings, TILE.BAS and TILEDEMO.BAS, from the *DOS World* BBS (603-924-3181). But the most economical and educational way to obtain copies of the programs is to type them in. Even if you've never touched a line of QBasic before, you should find it easy if you type carefully. (See "How to Use This Magazine," pages 70 to 72 in this issue, for help on using the *DOS World* BBS and typing in QBasic programs.)

First, load QBasic by typing the command C:\DOS\QBASIC at the DOS prompt. (Change the path if you store QBasic in a subdirectory other than C:\DOS.) Then press Esc to clear the opening screen.

To avoid losing any of your work, save TILE.BAS right away by choosing File/Save (or File/Save As) and typing its path and name in the File Name box.

For example, if you want to store the file in C:\QBASIC\PROGRAMS, type the following line into the box:

C:\QBASIC\PROGRAMS\TILE.BAS

As you work, remember to save the program periodically.

The next step is typing in the relatively short main program, which consists of a couple of DIM statements, screen-initialization commands, and a loop that scans the keyboard for input. To save time, omit the DECLARE statements. When you create and save the program's subroutines, QBasic adds those lines automatically.

Next, create the first subroutine, COLORBOX, by typing this line:

SUB COLORBOX
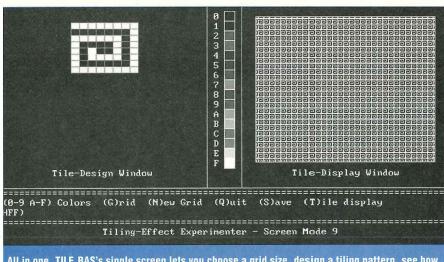
When you press Enter, QBasic will move you to another screen and display the following lines:

SUB COLORBOX
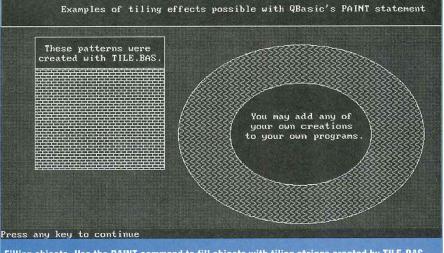
END SUB

Position your cursor on the blank line, then type the five remaining lines of the subroutine. This is probably a good time to save your work again.

To begin the next subroutine, COMMANDIN, press F2 to bring up the Subs dialog box; highlight the main program, TILE.BAS; press

All in one. TILE.BAS's single screen lets you choose a grid size, design a tiling pattern, see how that pattern looks when repeated, and save the command for creating your pattern to a file.

Filling objects. Use the PAINT command to fill objects with tiling strings created by TILE.BAS.

HELP WITH BOOKMARKS

Windows TIP

If you refer often to the same sections of Windows' Help utility, try using bookmarks. Placing a bookmark in a particular topic lets you revisit that topic with just four clicks of the mouse: on Help, on Contents, on Bookmark, and on the item you want to review. That's a lot easier than scrolling through Help's contents and trying to relocate a subject you consulted earlier.

Of course, before you can retrieve a Bookmark selection, you must store it. Start by opening Help and clicking on the topic in which you're interested. Next, click on Bookmark in the Help title bar, then on Define. The Bookmark window will open, and the title of the item under study will appear in the Bookmark Define box. (To rename the topic using a phrase you find more descriptive, simply type in a new heading.) Click on OK to earmark this topic for instant retrieval.

To delete a bookmark from the list of entries, highlight that item in the Bookmark window and click on Delete.

—Lane Olinghouse

Enter; and type SUB COMMANDIN. After typing in the subroutine's final LOOP statement, save your work again, then return to the main program as before.

Follow the same procedures for creating the remaining subroutines. If QBasic interrupts you by displaying an error message, crosscheck your typing against the commands shown in the listing, and fix your mistake.

When you finish typing, save the listing one more time, then press Shift+F5 to run the program. If you typed the code correctly, you'll see a display like the one in the first screen shot ("All in one," top).

TILE.BAS lets you design a pattern block using screen 9's 16 colors. When you view the wallpaper effect achieved by repeating this block, the program generates a tile string, which you may use to reproduce the effect in another QBasic program.

```
DECLARE SUB SAVEIT ()
DECLARE SUB NEWGRID ()
DECLARE SUB DISPLAY ()
DECLARE SUB COLORBOX ()
DECLARE SUB EDITCURS ()
DECLARE SUB COMMANDIN ()
DECLARE SUB SETCURS ()
DECLARE SUB GRID ()
DECLARE SUB SCREENINIT ()

'Begin typing here:
DIM SHARED BOXCOLOR(1 TO 8, 1 TO 16)
DIM SHARED BIT(1 TO 4)

GRIDSIZE = 8: CURSX = 1: CURSY = 1
XSTART = 98: YSTART = 28
OLDCURSX = 8: OLDCURSY = 8
BOXCOLOR = Ø
CLS
SCREEN 9
CALL SCREENINIT
DO
    CALL COMMANDIN
LOOP

SCREEN Ø
END   'End of main program

SUB COLORBOX
SHARED GRIDSIZE, CURSX, CURSY, XSTART, YSTART, COMMIN$
COLORTYPE = ASC(COMMIN$)
IF COLORTYPE >= 65 THEN COLORTYPE = COLORTYPE - 7
BOXCOLOR(CURSX, CURSY) = COLORTYPE - 48
CALL SETCURS

END SUB

SUB COMMANDIN
SHARED GRIDSIZE, CURSX, CURSY, XSTART, YSTART,
   COMMIN$, TILE$

DO
   DO
       COMMIN$ = INKEY$
   LOOP UNTIL COMMIN$ <> ""
       IF LEN(COMMIN$) = 2 THEN CALL EDITCURS
       IF ASC(COMMIN$) >= 48 AND ASC(COMMIN$)
          <= 57 THEN CALL COLORBOX
       COMMIN$ = UCASE$(COMMIN$)
       IF ASC(COMMIN$) >= 65 AND ASC(COMMIN$)
          <= 7Ø THEN CALL COLORBOX
       IF COMMIN$ = "Q" THEN SCREEN Ø: END
       IF COMMIN$ = "T" THEN CALL DISPLAY
       IF COMMIN$ = "G" THEN CALL NEWGRID
       IF COMMIN$ = "N" THEN CALL GRID
       IF COMMIN$ = "S" THEN CALL SAVEIT
LOOP

END SUB

SUB DISPLAY
SHARED GRIDSIZE, TILE$, TILETEXT$
LINE (363, 2Ø)-(617, 228), Ø, BF
```

```
LINE (363, 2Ø)-(617, 228), , B
TILETEXT$ = ""
TILE$ = ""
FOR Y = 1 TO GRIDSIZE
  BITECOUNT = 7
  FOR Z = 1 TO 4: BIT(Z) = Ø: NEXT Z
  FOR X = 1 TO 8
     BITE = BOXCOLOR(X, Y)
     FOR Z = 3 TO Ø STEP -1
        R = BITE \ 2 ^ Z
        IF R < 1 THEN R = Ø ELSE BITE = BITE - 2 ^ Z
        BIT(Z + 1) = BIT(Z + 1) + (R * 2 ^ BITECOUNT)
     NEXT Z
  BITECOUNT = BITECOUNT - 1
  NEXT X
TILETEXT$ = TILETEXT$ + "+ CHR$(&H" + HEX$(BIT(1))
   + ") + CHR$(&H" + HEX$(BIT(2)) + ") + CHR$(&H" +
   HEX$(BIT(3)) + ") + CHR$(&H" + HEX$(BIT(4)) + ")"
TILE$ = TILE$ + CHR$(BIT(1)) + CHR$(BIT(2)) +
   CHR$(BIT(3)) + CHR$(BIT(4))
NEXT Y
VIEW PRINT 22 TO 22
LOCATE 22, 1: PRINT SPACE$(79);
LOCATE 22, 1: PRINT "Tile="; TILETEXT$;
PAINT (4ØØ, 1ØØ), TILE$, 15

END SUB

SUB EDITCURS
SHARED GRIDSIZE, CURSX, CURSY, XSTART, YSTART,
   COMMIN$, OLDCURSX, OLDCURSY
CONST UPARROW% = 72
CONST DOWNARROW% = 8Ø
CONST RIGHTARROW% = 77
CONST LEFTARROW% = 75
CONST HOMEKEY% = 71
CONST ENDKEY% = 79
CONST PGUP% = 73
CONST PGDN% = 81

OLDCURSX = CURSX: OLDCURSY = CURSY

SELECT CASE ASC(RIGHT$(COMMIN$, 1))
   CASE UPARROW%
      CURSY = CURSY - 1
      IF CURSY <= Ø THEN CURSY = 1
   CASE DOWNARROW%
      CURSY = CURSY + 1
      IF CURSY >= GRIDSIZE THEN CURSY = GRIDSIZE
   CASE RIGHTARROW%
      CURSX = CURSX + 1
      IF CURSX >= 8 THEN CURSX = 8
   CASE LEFTARROW%
      CURSX = CURSX - 1
      IF CURSX <= Ø THEN CURSX = 1
   CASE HOMEKEY%
      IF (CURSX > 1) AND (CURSY > 1) THEN
         CURSX = CURSX - 1
         CURSY = CURSY - 1
      END IF
      REM IF CURSX <= Ø THEN CURSX = 1
      REM IF CURSY <= Ø THEN CURSY = 1
   CASE ENDKEY%
```

```
      IF (CURSX > 1) AND (CURSY < GRIDSIZE) THEN
         CURSX = CURSX - 1
         CURSY = CURSY + 1
      END IF
      REM IF CURSX <= 0 THEN CURSX = 1
      REM IF CURSY >= GRIDSIZE THEN CURSY = GRIDSIZE
   CASE PGUP%
      IF (CURSX < 8) AND (CURSY > 1) THEN
         CURSX = CURSX + 1
         CURSY = CURSY - 1
      END IF
      REM IF CURSX >= 8 THEN CURSX = 8
      REM IF CURSY <= 0 THEN CURSY = 1
   CASE PGDN%
      IF (CURSX < 8) AND (CURSY < GRIDSIZE) THEN
         CURSX = CURSX + 1
         CURSY = CURSY + 1
      END IF
      REM IF CURSX >= 8 THEN CURSX = 8
      REM IF CURSY >= GRIDSIZE THEN CURSY = GRIDSIZE

END SELECT

CALL SETCURS

END SUB

SUB GRID
SHARED GRIDSIZE, CURSX, CURSY, XSTART, YSTART
FOR X = 1 TO 8
   FOR Y = 1 TO 16
      BOXCOLOR(X, Y) = 0
   NEXT Y
NEXT X

FOR Y = YSTART TO (9 * 16 - 1) + YSTART STEP 9
   Y1 = Y
      FOR X = XSTART TO 12 * 7 + XSTART STEP 12
         LINE (X, Y1)-(X + 12, Y1 + 9), 0, BF
```

```
      COUNT = COUNT + 1
      NEXT X
NEXT Y

FOR Y = YSTART TO (9 * GRIDSIZE - 1) + YSTART STEP 9
   Y1 = Y
      FOR X = XSTART TO 12 * 7 + XSTART STEP 12
         LINE (X, Y1)-(X + 12, Y1 + 9), 7, B
      COUNT = COUNT + 1
      NEXT X
NEXT Y
CALL SETCURS
END SUB

SUB NEWGRID
SHARED GRIDSIZE, CURSX, CURSY, XSTART, YSTART, COMMIN$
DO
   LOCATE 22, 1
   PRINT SPACE$(79);
   LOCATE 22, 1
   INPUT ; "Enter new gridsize (8, 12 or 16) ",
       GRIDSIZE
LOOP UNTIL GRIDSIZE = 8 OR GRIDSIZE = 12
   OR GRIDSIZE = 16
LOCATE 22, 1
PRINT SPACE$(79);
LOCATE 22, 1
PRINT "No messages";
CURSX = 1: CURSY = 1
CALL GRID

END SUB

SUB SAVEIT
SHARED TILE$, TILETEXT$
LOCATE 22, 1
PRINT SPACE$(79);
LOCATE 22, 1
INPUT ; "Enter path and filename "; TILENAME$
```

You may go on to create TILE-DEMO.BAS at this point, or linger awhile and tinker with TILE.BAS.

By Design

TILE.BAS's program screen contains four main elements: a tile-design window, where you create your pattern; a color bar, which indicates your color options; a tile-display window, which shows how your pattern will look when tiled; and a command and message area.

When you start the program, the cursor is in the upper-left corner of the tile-design grid, so that you can begin designing a pattern block immediately. To move the cursor left, right, up, or down, press the arrow keys; to move diagonally, use Home, Page Up, End, or Page Down. To mark a square with a particular color, position the cursor, then press the number or letter corresponding to the color you want. For instance, to color a square white, press F.

When you're ready to see what sort of wallpaper effect your tile produces when repeated, press the T key. The screen's message line displays the tile string used to create that effect. (If the string is long, only its final characters are visible.)

When you edit the tile pattern and press the T key to update the tile-display window, the tile string changes. To clear the existing pattern, press the N key to display a new grid. By default, the design grid is 8 pixels by 8 pixels. To choose an 8-by-12-pixel grid or an 8-by-16-pixel grid, press the G key.)

When you're satisfied with your tile, save it to disk by pressing the S key. The program prompts you for a path and filename.

For instance, to save a file called FRET and store it in C:\QBASIC\TILE, type this line:

C:\QBASIC\TILE\FRET

Laying Tiles

After creating a few tile strings, you're ready to begin filling objects with them. To understand the process, take a closer look at TILE-

```
IF TILENAME$ = "" THEN LOCATE 22, 1: PRINT
    SPACE$(79); : EXIT SUB
TILETEXT$ = "TILE$ = " + TILETEXT$

OPEN TILENAME$ FOR OUTPUT AS #1
PRINT #1, TILETEXT$
CLOSE
LOCATE 22, 1
PRINT SPACE$(79);
LOCATE 22, 1
PRINT "Tile design has been saved as "; TILENAME$

END SUB


SUB SCREENINIT
SHARED GRIDSIZE, CURSX, CURSY, XSTART, YSTART
REM SETUP SCREEN TEXT
FOR X = Ø TO 15
    X$ = STR$(X)
    LOCATE X + 2, 39
    PRINT HEX$(X);
NEXT X
LOCATE 18, 1Ø: PRINT "Tile-Design Window"; SPACE$(25);
    "Tile-Display Window";

VIEW PRINT 23 TO 24
TEXT$ = "Tiling-Effect Experimenter - Screen Mode 9"
PRINT STRING$(8Ø, "=")
PRINT TAB(4Ø - (LEN(TEXT$) / 2)); TEXT$;
VIEW PRINT 20 TO 22
PRINT STRING$(8Ø, "=")
LOCATE 21, 1: PRINT "(Ø-9 A-F) Colors  (G)rid
    (N)ew Grid  (Q)uit  (S)ave  (T)ile display";
LOCATE 22, 1
PRINT "No messages";

'Set up view port
VIEW (1, 1)-(638, 265), , 15
```

```
'Set up view-port windows
LINE (293, Ø)-(343, 265), , B

'Set up up color bar
LINE (318, 12)-(333, 237), , B
LINY = 12
FOR X = 1 TO 16
    LINY = LINY + 14
    LINE (318, LINY)-(333, LINY)
    PAINT (32Ø, LINY - 2), X - 1, 15
NEXT X

'Set up Tile-Display Window
LINE (363, 20)-(617, 228), , B

'Set up Tile-Design Window
CALL GRID

END SUB

SUB SETCURS
SHARED GRIDSIZE, CURSX, CURSY, XSTART, YSTART,
    OLDCURSX, OLDCURSY, BOXCOLOR

LINE (OLDCURSX * 12 + XSTART - 12, OLDCURSY * 9 +
    YSTART - 9)-(OLDCURSX * 12 + XSTART, OLDCURSY *
    9 + YSTART), 7, B
LINE (CURSX * 12 + XSTART - 12, CURSY * 9 + YSTART
    - 9)-(CURSX * 12 + XSTART, CURSY * 9 + YSTART),
    BOXCOLOR(CURSX, CURSY), BF

LINE (CURSX * 12 + XSTART - 12, CURSY * 9 + YSTART
    - 9)-(CURSX * 12 + XSTART, CURSY * 9 + YSTART),
    15, B

END SUB
```

End

DEMO.BAS. It starts by clearing the screen and issuing a SCREEN 9 statement, the same screen TILE.BAS uses when creating tile strings. (If you prefer another QBasic screen, you must modify the tiling formula in the DISPLAY subroutine.)

Next come the demo's three tiling strings: TILE1$, the blue-and-gray background; TILE2$, the red-brick pattern; and TILE3$, the squiggle pattern.

The LINE and CIRCLE statements in the next section of the program draw the objects. In QBasic, the fundamental syntax for the outline of a box is as follows:

```
LINE (x1, x2)-(y1, y2), c, b
```

where *x1* is the distance (in pixels) from the left edge of the screen to the upper-left corner of the box; *x2* is the distance from the top of the screen; *y1* is the distance from the right edge of the screen; *y2* is the distance from the bottom of the screen; the letter *c* represents a color value from zero to 15; and *b* indicates that you're drawing a box rather than a line.

For screen 9, the display's resolution is 640 pixels across by 350 pixels down. The first screen location is (0,0), so the highest permissible *x* value is 639, and the highest *y* value is 349.

For a CIRCLE statement, the basic syntax is as follows:

```
CIRCLE (x, y), r, bc
```

where (*x*, *y*) provides the coordinates of the center point; *r* is the radius in pixels; and *bc* is the color value of the border.

The first LINE statement in TILEDEMO.BAS draws the box at the top of the screen. The next two LINE statements superimpose a large box on a smaller box. The smaller, top box holds explanatory text, and the bottom one holds the red-brick pattern. The two CIRCLE statements create two concentric circles; the doughnut shape holds the pattern; and the hole contains the text.

With the screen chosen, tiling strings defined, and object and text

TILEDEMO.BAS shows how you may incorporate the tile strings created by TILE.BAS into your own programs. If you type in the tile strings manually (rather than copying them from a file created by TILE.BAS), be sure to type each string as a single line.

```
CLS
SCREEN 9

'Blue/gray background screen
TILE1$ = CHR$(&HAA) + CHR$(&HAA) + CHR$(&HAA) + CHR$(&H0) +
  CHR$(&H55) + CHR$(&H0) + CHR$(&H0) + CHR$(&H0) + CHR$(&HAA) +
  CHR$(&HAA) + CHR$(&HAA) + CHR$(&H0) + CHR$(&H55) + CHR$(&H0) +
  CHR$(&H0) + CHR$(&H0) + CHR$(&HAA) + CHR$(&HAA) + CHR$(&HAA) +
  CHR$(&H0) + CHR$(&H55) + CHR$(&H0) + CHR$(&H0) + CHR$(&H0) +
  CHR$(&HAA) + CHR$(&HAA) + CHR$(&HAA) + CHR$(&H0) + CHR$(&H55) +
  CHR$(&H0) + CHR$(&H0) + CHR$(&H0)

'Red-brick pattern
TILE2$ = CHR$(&H4) + CHR$(&H4) + CHR$(&HFF) + CHR$(&H4) +
  CHR$(&H4) + CHR$(&H4) + CHR$(&HFF) + CHR$(&H4) + CHR$(&H4) +
  CHR$(&H4) + CHR$(&HFF) + CHR$(&H4) + CHR$(&HFF) + CHR$(&HFF) +
  CHR$(&HFF) + CHR$(&HFF) + CHR$(&H20) + CHR$(&H20) + CHR$(&HFF) +
  CHR$(&H20) + CHR$(&H20) + CHR$(&H20) + CHR$(&HFF) + CHR$(&H20) +
  CHR$(&H20) + CHR$(&H20) + CHR$(&HFF) + CHR$(&H20) + CHR$(&HFF) +
  CHR$(&HFF) + CHR$(&HFF) + CHR$(&HFF)

'Squiggle pattern
TILE3$ = CHR$(&HF0) + CHR$(&H30) + CHR$(&HF0) + CHR$(&HC0) +
  CHR$(&HFC) + CHR$(&HC) + CHR$(&H3C) + CHR$(&H30) + CHR$(&H3F) +
  CHR$(&H3) + CHR$(&HF) + CHR$(&HC) + CHR$(&HF) + CHR$(&H0) +
  CHR$(&H3) + CHR$(&H3) + CHR$(&HF) + CHR$(&HC) + CHR$(&HF) +
  CHR$(&H3) + CHR$(&H3F) + CHR$(&H30) + CHR$(&H3C) + CHR$(&HC) +
  CHR$(&HFC) + CHR$(&HC0) + CHR$(&HF0) + CHR$(&H30) + CHR$(&HF0) +
  CHR$(&H0) + CHR$(&HC0) + CHR$(&HC0)

LINE (0, 0)-(639, 45), 15, B
LOCATE 2, 12
PRINT "Examples of tiling effects possible with QBasic's PAINT statement";
LOCATE 6, 9
PRINT "These patterns were";
LOCATE 7, 8
PRINT "created with TILE.BAS.";
LOCATE 13, 46
PRINT "You may add any of";
LOCATE 14, 46
PRINT "your own creations";
LOCATE 15, 45
PRINT "to your own programs.";
LINE (50, 60)-(235, 250), 15, B
LINE (50, 60)-(235, 105), 15, B
CIRCLE (430, 200), 175, 15

CIRCLE (430, 200), 100, 15

'Paint the background screen
PAINT (638, 300), TILE1$

'Paint the red bricks
PAINT (200, 200), TILE2$

'Paint the squiggles
PAINT (535, 300), TILE3$
```

End

in place, only one task remains: painting in the tiling strings. As I mentioned earlier, the PAINT statement handles this task. Here's the appropriate syntax:

$$PAINT\ (x, y),\ s,\ c$$

where (x, y) provides the screen coordinates of the point at which you want to start painting; the letter s represents the name of the tiling string you want to use; and c is the number of the border color. (You may omit that value, as I did in TILEDEMO.BAS.)

The starting point for painting may name any x and y values that fall within the object you're painting; QBasic paints the tiling string right up to the object's boundaries. Of course, for QBasic to paint a tiling string, the string must precede the PAINT statement; that's why I placed TILE1$, TILE2$, and TILE3$ near the top of the program.

So far so good, but what if you want to substitute a different tiling string for TILEDEMO.BAS's dotted blue-and-gray background? That's no problem:

● Load the listing into an editor that saves files in ASCII (text) format and offers a merge function (or lets you copy and paste between documents).
● Then delete the existing tiling string and merge the new one with the listing.
● Save the program under a different name—perhaps TILEDEM2.BAS—and then run the new listing.

What's next? That's entirely up to you. I do have one bit of advice, though: Have fun. Tiling opens up so many possibilities for improving the look of your QBasic programs that you owe yourself at least a couple of hours of play time.

Start with something simple, such as your initials, and then branch out to planets, faces, light bulbs, or, better yet, how about a bolt of lightning? ■

START-UP CLINIC

In the Beginning There Was Trouble

by Jack Nimersheim

This month's letters to "Start-Up Clinic" are somewhat unique in that they all deal with issues related to the process of getting started—literal "start-up" problems or questions, if you will. A lot happens when you first turn on your computer or load a program. Some procedures you can modify; some you can't. Knowing your start-up options will help you understand how you can tweak your computer to work more efficiently.

Understanding the basics of CMOS, IRQs, and password protection can help you survive a host of nagging start-up problems.

Under the Hood

Whenever I turn on my computer, it displays the following message: "Press <Del> if you want to run SETUP / EXTD set-up." What does that line mean and how can I use the resulting information?

Moe Lynn
Edmonton, Alberta
Canada

During start-up, your computer gives you the opportunity to access and modify its CMOS settings. Basically, these settings, which are stored on a *complementary metal-oxide semiconductor* chip, specify your hardware's setup. This information lets your system start successfully. (Computer engineers use a CMOS chip because, unlike standard memory chips, it can retain its settings by means of a low-power battery, even when your computer is turned off.) The settings stored in CMOS cover a variety of items, including the date and time, your video-adapter type, how many and what kinds of floppy disks are installed, and the size and setup of your hard disk.

Some CMOS settings are provided merely for convenience. Others are necessary for your computer to run properly. If the date and time are set incorrectly, for example, your system will work reliably, but a file's date/time stamp won't reflect accurately the last time you modified it. The hard-disk settings, on the other hand, are critical. Much like a roadmap, they let your computer's BIOS (*basic input / output system*) locate the sectors and tracks your system is using to store and retrieve the data on your hard disk—including the operational core of MS-DOS, the files that load into memory during system start-up.

The content and structure of the CMOS setup routine vary slightly from system to system, depending on the type of BIOS it contains. My 486's AMI BIOS, for example, offers two levels of CMOS setup—Standard and Advanced. All the items mentioned above are available through the Standard setup. The Advanced routine provides access to additional items, such as initial CPU speed, video cache memory, and the option of moving certain BIOS operations into RAM for increased system efficiency.

Here's some wise advice: If you don't know what you're doing, don't mess with your computer's CMOS settings. An improper setting can wreak havoc with your PC, possibly even stop it from booting successfully and advancing to the DOS system prompt. If you're curious about your PC, however, find a friend who understands CMOS settings and go through them together; just make sure you make a copy of your CMOS settings in case something goes wrong. By taking the time to explore, you can learn a lot about your computer's components and how they work.

For an in-depth discussion of CMOS, see "Without You I'm Nothing," page 28 in this issue.

IRQsome Interrupts

I've encountered problems with certain shareware sequencer programs and my Sound Blaster 16 sound card. I believe the difficulties may be caused by an interrupt conflict. According to the article "Solving IRQsome Problems," which appeared in DOS World *#17 [September 1994, page 26], IRQs 10 through 12 are reserved for future expansion. When I run* Microsoft Diagnostics *(MSD) on my system, however, it indicates that the BIOS is using IRQ 10. Which is it? Can I use IRQ 10 for my sound card? And will that resolve my conflict problems?*

Eugene Brandon
Los Angeles, Calif.

Ken Johnson's article is correct. When the IBM "standard" was established, interrupt 10 (IRQ 10) was reserved for "future operations." To date, those future operations haven't materialized. (As the article's chart points out, IRQ 12 was also initially reserved for future use. On PS/2 systems, it has since been designated as the IRQ that manages the mouse port.) Unless you've already set up an add-on board to use IRQ 10, therefore, it should be available now for your sound card.

Your confusion stems from the ambiguous terms MSD uses to display information about IRQ status. I'd guess that nothing is listed for IRQ 10 under the heading "Detected" in the MSD display. Rather, the column is probably blank, indicating that MSD detected no device currently set up to use that interrupt.

The BIOS reference you mention probably appears in the far-right column, marked "Handled By." All it denotes is that control of IRQ 10 is being handled through the BIOS, the normal condition when no other device driver has superseded the BIOS's control over that interrupt.

Try setting up your sound board and its software to use IRQ 10. Then reboot your system and rerun MSD. At this point, the values listed for IRQ 10 should change, because your sound card and its device driver are now using it.

Safeguarding Your System

Can I incorporate into my start-up routine a batch program that executes before my CONFIG.SYS *file? I want to password-protect my system. Would such a batch file let me do that?*

Brian R. Kaufman
New York, N.Y.

Sorry, Brian, but the approach you propose won't work for two reasons. First, DOS doesn't let you execute a batch program before CONFIG.SYS; second, a knowledgeable user can easily defeat any protection scheme based exclusively on the contents of CONFIG-

.SYS or AUTOEXEC.BAT—or any batch file, for that matter. How easily? In MS-DOS 5 and later, all someone has to do is press the F5 function key when the "Starting MS-DOS" message appears. F5 instructs DOS to bypass the CONFIG.SYS and AUTOEXEC.BAT files and advance directly to the DOS system prompt. DOS 6.0 and later versions offer the additional option of using the F8 function key to execute selected individual commands in your CONFIG.SYS and AUTOEXEC.BAT files.

Having said that, however, here's a possible alternative. Your computer may contain a built-in password feature, although you may not realize it. The key to discovering whether it does lies in the CMOS settings discussed in "Under the Hood," page 63.

One of the options available on my 486's CMOS menu, for example, is Change Password. Initially—that is, when my system was shipped—the manufacturer left that value blank. By default, lack of a password told my system to bypass password security and immediately begin executing the contents of CONFIG.SYS. It was a practical approach, in that it let me turn on the computer and begin using it without having to know a specific password.

Indeed, I wasn't even aware that my new system offered password protection until I read about that feature in the documentation. One of the first things I did after trying out my system, therefore, was to run the CMOS setup program and use the Change Password option to specify a password. Doing so automatically activated password protection for my system. Now whenever someone turns on my computer, he or she must type in the password I've specified or the system won't boot, which denies that person access to my programs and files.

You might want to check your system manuals to see whether your computer offers the same feature. If you don't have the manuals or can't find them, try pressing the Delete key to access CMOS setup during start-up, as described above (although that doesn't work on all computers; some use other keys or key combinations, such as Ctrl+Alt+S, for example). You may be pleasantly surprised to discover a password option listed on its CMOS menu.

Even if password protection isn't "hard-wired" into your system, don't despair. Several commercial programs include password protection among their features. As a rule, these add-on schemes safeguard your system by locking the keyboard and screen until someone enters the required password. Symantec's popular suite, The Norton Utilities, is one example of software that includes a password-protection option. ■

Jack Nimersheim has been writing about computers for 14 years. He's the author of two dozen books on computer-related topics. His newest work, Windows95 Revealed *(Random House Electronic Publishing), was released last fall.*

WINDOWS SEAT

Automatic Windows

by Tony Roberts

For most hard-core DOS users, batch files are an essential part of life; they make computing chores faster and more efficient. Many of these DOS users who begin using Windows are disappointed that Windows gives them nothing to match the power of DOS batch files. Or so they think.

Windows' Program Manager can perform some of the same tasks batch files do. For instance, one commonly used batch file switches to an application's subdirectory, then loads the application. This makes the application's subdirectory the default, and files created by the application are stored there.

The same task is simple in Program Manager. Click once on any Program Manager icon to select it, and then select File/Properties. The dialog box that's displayed shows the command line required to execute the program and also offers space for you to specify a working directory if you'd prefer to store your data files in a location separate from the program files. Thereafter, a double click on the program icon in Windows has the same effect as a DOS batch file.

For batch-file tasks within Windows applications, experiment with the Recorder applet. (If you're using Windows' default groups, it's in Accessories.) Recorder captures keystrokes and mouse movements and lets you play them back later by pressing a key combination.

At first glance, using Recorder can be so frustrating that you may be tempted to delete the program entirely. But don't be too hasty. Once you understand what Recorder is—and what it isn't—you can make it work for you.

> *In essence, Recorder macros customize your applications so that they work the way you want them to.*
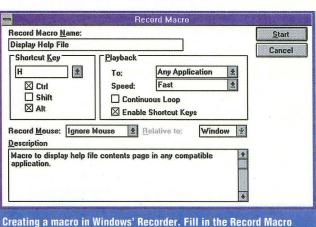
Simplifying Windows Tasks

Recorder can save you keystrokes and automate tasks you do repeatedly within Windows. For example, you can use a Recorder macro from within File Manager to open an ASCII file, such as a system configuration file or a README file, for viewing or editing. (See the accompanying sidebar "Creating a Windows File Viewer," page 66, for step-by-step instructions.) Or you can create macros to alter the sorting of File Manager's directory display. And if you repeatedly make the same selections from a dialog box such as Print or Page Setup, you might want to create a macro that does the work for you.

In a word-processing or desktop-publishing program, you can use a Recorder macro to apply formatting to selected text. If your telecommunications soft-

Creating a macro in Windows' Recorder. Fill in the Record Macro dialog box as shown above to create a macro that will display the contents of a Help file.

ware doesn't offer macros of its own, use Recorder to move quickly to your on-line destinations. In essence, macros customize your applications so that they work the way you want them to.

Working with Macros

To get started working with macros, let's begin with a quick one that will display the Help file for the current application. Double-click on the Recorder icon. Select Macro/Record and fill in the dialog as illustrated in the accompanying screen shot, "Creating a macro in Windows' Recorder" (page 65).

Enter a descriptive name (you can use several words) for the macro. Assign a shortcut key combination— let's use Ctrl+Alt+H. In the Playback section, select Any Application and Fast Speed. Select Ignore Mouse as the Record Mouse option. Use the description box at the bottom of the dialog box to record information about how the application should use the macro. In this case, it's evident, but as your macros get more complex, you'll appreciate having a set of instructions.

Although Recorder can capture mouse movements as well as keystrokes, in general it's best to select Ignore Mouse. The reason is that your mouse movements won't play back correctly unless your windows

CREATING A WINDOWS FILE VIEWER

Get to know Windows' Recorder by trying out this sample macro. When this macro is running, highlight any file in File Manager, press the shortcut key for the macro, and the file will be displayed as a Notepad file.

1. Open File Manager on your desktop.
2. Open Recorder and select Macro/Record. Enter a name and shortcut key for the macro.
3. Under Playback, select Same Application.
4. Make sure Ignore Mouse is selected.
5. Click on Start to begin recording.
6. Use the mouse to select File Manager and to select a target file—for example, CONFIG.SYS.
7. Press Alt+F, R to start the Run command. The filename CONFIG.SYS will appear selected in the Command Line box.
8. Press Ctrl+Insert to copy the selected filename to the Windows Clipboard.
9. Type the word NOTEPAD followed by a space.
10. Press Shift+Insert to retrieve the target filename from the Windows Clipboard.
11. Press Enter to execute the command. (The OK box in the Run dialog is the default, so it executes when you press Enter.)
12. Click on the blinking Recorder icon (or press Ctrl+Break) and save the macro.

—T.R.

are set up precisely the way they were during recording. Because it's almost impossible to ensure perfect placement of elements on your screen, it's much safer to record macros using keyboard commands only.

We're now ready to record the macro. Select Start, and Recorder will collapse into a minimized blinking icon. This indicates that Recorder is at work. Use your mouse (remember, the software isn't recording mouse movements) to select a Windows application such as Program Manager or File Manager. Now, using the keyboard, press Alt+H to activate the Help menu, then press C to select Contents. The current application's Help file should load and display the contents page.

To complete the process, use the mouse to activate Recorder or press Ctrl+Break; select Save Macro and click on OK. The macro has been recorded. Now when the macro is running and you press Ctrl+Alt+H from within most applications, you'll have instant access to the Help file.

I say "most applications" because conflicts will arise when an application's own shortcut keys are identical to the key combinations you've assigned. To circumvent that

HOW FILE MANAGER (SOMETIMES) BEATS DOS

Windows TIP

Windows' File Manager does a few things better than DOS. It copies disks faster, because it reads all of the disk into memory and then does its copying. It copies files more efficiently, too, because when a disk fills and you get a "Disk full" message, you may insert a fresh disk and keep on copying.

It's easier to copy, move, and delete files in File Manager than from the DOS command line, because you don't have to figure out a complicated wildcard pattern. The filenames are listed on screen, and you may press Ctrl and click the left mouse button to highlight multiple files. Telling Windows what to do with the selected files is simple, too: Press F8 to copy files, F7 to move them, and, of course, Del to delete them.

Renaming and deleting directories also is simpler in File Manager than in DOS, unless you've upgraded to DOS 6.x, which offers the MOVE and DELTREE commands. (These commands let you relocate and rename files and directories, and delete directories that aren't empty.) And, to reorganize directories, you may drag them to a new position in the directory tree.

—Kay Yarborough Nelson

problem, activate Recorder, select your macro, then choose Properties. From the Properties dialog box, assign a new shortcut key.

Before we go further, use the Save As command to save your macro. You can store all your macros in one file or keep them in separate files, which you can load as needed. I store most of my macros (a couple of dozen) in a single file called MACROS.REC. I start Recorder and load all my macros as a matter of course each time Windows starts. To load your Recorder file automatically when Windows starts, select the Recorder icon in Program Manager and press F8 to copy it to the StartUp group. Then open the StartUp group, click on the Recorder icon and select File/Properties. Replace the command line with the following line:

```
C:\WINDOWS\RECORDER.EXE
C:\WINDOWS\MACROS.REC
```

Replace MACROS.REC with the name of your macro file. Also click on Run Minimized so that the macro will appear as an icon at the bottom of your screen.

Get It Right—Or Else

One of the biggest drawbacks of Recorder macros is that you can't edit them. You can change the properties of a macro—such as the shortcut key combination—but you can't alter the actual keystrokes. If you find that you've made a mistake or need to add a new wrinkle to your macro, you'll have to delete the old macro and record a new one from scratch.

You can view the contents of a macro, however, by selecting the macro and pressing and holding down the Shift key while selecting Macro/ Properties. This undocumented command lists everything that happened while you recorded the macro keypress by keypress. You'll see two entries for each keypress: one as you pressed the key down and another as you released it.

Another drawback of Recorder macros is that they work only in Windows applications. You can't create macros to work with DOS applications running under Windows. (For a shareware alternative to Recorder—one that does work

with both Windows and DOS—see the accompanying sidebar "More Power for Windows," below.)

Be on the Lookout

Despite these limitations, however, Recorder macros can improve your control over Windows programs and save you time. Look for repetitious tasks as you work and use Recorder to automate them. With a little investigation, you'll find several procedures for which you can collapse many clicks or keystrokes into a single key combination. ∎

Tony Roberts owns and operates a desktop-publishing business and has been writing about personal computers for 12 years.

MORE POWER FOR WINDOWS

If you're serious about automating your Windows work, take a look at WinBatch, a shareware offering from Wilson WindowWare. WinBatch is more like the familiar DOS batch language than Recorder is, but that's just the beginning. WinBatch lets you launch and control both Windows and DOS applications running serially or simultaneously.

To work with WinBatch, you use the Windows Interface Language (WIL) to create scripts in a text editor such as Notepad. WIL offers more than 350 functions and commands that let you launch programs, pass keystrokes to applications, request input from the user, and act upon that input.

The ingenious sample/tutorial included with WinBatch shows you how to write a script to run the Windows Solitaire program with a timer installed so that you don't let your gaming habit eat up your work day. When you run the script, the software asks how long you want to play. When your time is up, you get one chance to extend your play time, but after that, it's "back to work."

WinBatch includes a full complement of program-control statements (FOR, WHILE, SWITCH, and GOSUB) that let you anticipate and program for widely varying circumstances. The program also includes a dialog editor with which you can create an attractive interface for your scripts.

Whether you want to create a script to launch your telecommunications software and fetch your mail or to process data in a DOS application, with a little programming WinBatch will be up to the task. The program, now in version 5, is well seasoned. The latest version's new commands eliminate some of the kinks of previous versions, and execution time is improved, as well.

Evaluation copies are available from most shareware dealers and on-line services. Or order directly from Wilson WindowWare, 2701 California Ave. S.W., #212, Seattle, WA 98116, 800-762-8383 ($69.95).

—T.R.

The Control Attributes box is part of WinBatch's Dialog Editor. By combining selections from this menu, you build your own dialog boxes for use with your WinBatch scripts.

UPGRADE UPDATE

■ **Novell DR Multiuser DOS Lite,** a single-user version of Concurrent Controls' corporate multitasking DOS operating system, lets individual PC users run up to eight networked applications simultaneously, with all eight supporting full VGA graphics. You can also link it into a peer-to-peer network to allow shared access to programs. *Novell DR Multiuser DOS Lite, $295; Concurrent Controls Inc., 880 Dubuque Ave., South San Francisco, CA 94080; 415-873-6240; fax 415-873-6091; BBS 415-873-6256.*

■ **PCRESC-U version 2.0** is a disaster-recovery software utility designed to help overcome hard-disk crash. Version 2.0 creates and saves to disk a backup of all vital information on your hard disk, then uses that information to restore any damaged data resulting from a disk crash. *PCRESC-U 2.0, $39; EliaShim Microcomputers, 4005 Wedgemere Drive, Tampa, FL 33610; 813-744-5177; fax 813-744-5197.*

■ *PowerBasic 3.1* is an upgrade for all QuickBasic and PowerBasic owners, designed to offer improved compatibility with QuickBasic's programming dialect. With the newest version, PowerBasic programmers can have total control to define C-style functions, both within their Basic code and in external libraries. Other enhancements include improved code optimization, larger source files, expanded on-line help, and free source-code libraries and utilities. *PowerBasic 3.1, $149; upgrades from 3.0 $19.95, from Quick-Basic $99; PowerBasic Inc., 316 Mid Valley Center, Carmel, CA 93923; 800-780-7707; fax 408-659-8008.*

BUTTONS FOR DOS 5.0

DOS Makeover

Point-and-click your way through your favorite DOS programs, plus get a choice about how to boot up, with Buttons for DOS 5.0's graphical menu system.

by Robert Gryphon
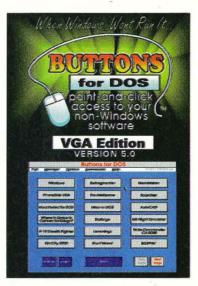
If you're primarily a Windows user, accustomed to launching programs at a click of a mouse button, typing DOS commands to run programs can be tedious—if not downright mystifying. A logical solution is to install software that will put your applications into menus for easy launching. Unfortunately, most menu systems are character-based, with no graphics and sometimes no mouse support, either.

Enter Triad Software's Buttons for DOS 5.0, a $29 graphical menu system for DOS applications, designed to look—if not act—like Windows.

Buttons for DOS's main interface consists of a title bar, a set of pull-down menus (for exiting, defining the applications you want to start, and configuring your system), three columns of panel-like application launch buttons that get you into your DOS programs, and a digital clock.

You can build up to ten pages of launch buttons—room for about 180 programs—and scroll through each one with your Page Up/Page Down keys. You can also create batch files and attach them to launch buttons with a "macro function."

When Buttons runs a DOS program, it removes virtually every byte of itself from memory to make more RAM available to your application. This process takes time, but it's certainly worth the wait.

A key enhancement in version 5.0 is Buttons' ability to define an "alternate bootup configuration." In simpler terms, that means you can create two versions of your start-up files (CONFIG.SYS and

AUTOEXEC.BAT), which you'll use for different purposes. For example, you might want one version to load memory-hungry network drivers and a CD-ROM driver for business purposes, but another when you need extra memory to run your computer games up to speed.

Fortunately, this process is mostly automated. When you choose the Alternate Startup Files option for one or more of your DOS applications, Buttons will look for drivers you can remove to save memory. Then when you launch the program or game for which you've checked the Alternate Startup Files option, your computer will reboot, reflecting the trimmed-down CONFIG.SYS and AUTOEXEC.BAT. Significantly, however, this feature is incompatible with DOS 6.x's multi-configurations.

Several other program flaws reduce Buttons for DOS's appeal. For example, the portion of the installation that modifies the AUTOEXEC.BAT file (which you can run at any time from within the main Buttons program)

features a dialog box in which the prompts blatantly overrun the frame. This function did the usual thing: modified my AUTOEXEC.BAT in an uninspired fashion, placing itself at the end of the file. I had to change it around a little afterwards to put Buttons in the right order in the batch file.

Also during installation, the program scanned for "popular" applications and placed any it found—including Windows programs—in menus. I don't have many DOS-only apps installed; those I had were identified correctly, but a number of EXEs in my Windows tree were identified incorrectly.

Although these problems detract from the software's overall intent, Buttons for DOS 5.0 is an inexpensive, easy alternative for those who don't want to type in DOS commands to run their favorite programs.

Robert Gryphon is a free-lance writer and programmer/

HOT FAX
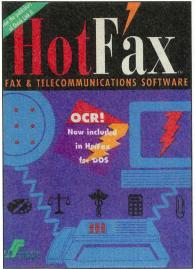
Getting the Message

Even if you all you've got is an ancient 8088 PC, you can still send and receive professional-quality faxes with this sizzling DOS-based communications program.

by Stanley J. Wszola

Hot Fax, a general-purpose data/fax communications package from Smith Micro Software Inc., can transform a DOS-based PC into a fax workstation. Even if you have an older PC, you can drop in an inexpensive fax/modem, add Hot Fax, and wind up with the near-equivalent of an expensive fax machine.

Hot Fax lets you send faxes to or receive faxes from any Group III fax machine or fax/modem. You can use the built-in text editor and cover sheet, or you can import text and graphics (charts, drawings, or photos) in various formats to create your own cover sheet and message. Hot Fax also supports several hand-held and flat-bed scanners.

A variety of fonts and sizes, in roman, boldface, and italic styles, provide quality printouts. Unfortunately, although you can enter embedded formatting commands into a text file, you can't see changes in style and size until you print. Also, the text editor isn't WYSIWYG; it displays text only in an 80-character by 25-line screen.

On the plus side, Hot Fax lets you store, search, or edit the names and numbers of fax recipients in one or more phone books (compatible with popular database programs), then categorize book entries into groups. You can program Hot Fax to schedule faxes for specified dates and times to a group of recipients, and to retransmit earlier faxes to a group.

A convenient feature lets you send faxes from within another DOS application, such as your word processor. Fast Fax V, a TSR, inter-

cepts output to your printer and reroutes it to the fax/modem. Hot Fax can receive faxes automatically; you can also set it to wait in the background and handle incoming faxes, so that you can keep working in another application, or to print automatically.

Hot Fax's hottest feature is its OCR (optical character recognition) capabilities: translating a graphics fax file into ASCII text. Accuracy depends on low-noise phone lines, a good-quality fax/modem, and the font style of the original.

Hot Fax is designed for DOS, but can run under Windows. The package includes both PIF and icon files, but running under Windows precludes background operations and OCR.

Stanley J. Wszola is a free-lance writer and systems-support analyst. Contact him on CompuServe at 71011,1726.

THE CONFIG.SYS FILE

In your root directory is a file called CONFIG.SYS. Like AUTOEXEC.BAT, this file is in ASCII, and you can view your CONFIG.SYS file with the TYPE command. A typical CONFIG.SYS might look like this:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH,UMB
FILES=50
BUFFERS=10
SHELL=C:\DOS\COMMAND.COM
  C:\DOS\ /E:1024 /P
DEVICE=C:\DOS\ANSI.SYS
DEVICE=C:\DOS\SETVER.EXE
```

The rules for handling CONFIG.SYS are the same as they are for AUTOEXEC.BAT: Always back up the original file before you modify it and always have an emergency boot disk available. As with AUTOEXEC.BAT, changes you make to CONFIG.SYS won't take effect until you restart your computer.

ANSI.SYS AND THE ESCAPE CHARACTER

When an article says you must have ANSI.SYS installed, it means that the MS-DOS file ANSI.SYS should be in your \DOS directory, and the following line should be in your CONFIG.SYS file:

```
DEVICE=C:\DOS\ANSI.SYS
```

Some articles that discuss ANSI.SYS will also ask you to create a batch file that uses the escape character. Unfortunately, there's no uniform method of doing so. If you use EDIT, the text editor that comes with MS-DOS, you can make an escape character by pressing Ctrl+P and then the Esc key. The escape character appears on screen as a small left-pointing arrow. If you're using another text editor or word processor, check its instructions for information on how to enter the escape character.

How to Use

THE AUTOEXEC. BAT FILE

Most people have a batch file called AUTOEXEC.BAT on their hard disk. If you want to look at it, first go to your root directory by typing CD\. Type DIR to make sure AUTOEXEC.BAT is there. Then type the following command:

```
TYPE AUTOEXEC.BAT | MORE
```

A simple AUTOEXEC.BAT file might look like this:

```
@ECHO OFF
PROMPT $P$G
PATH=C:\DOS;C:\WINDOWS;C:\WP51;C:\BAT
C:\DOS\SMARTDRV.EXE
C:\MOUSE\MOUSE.COM
C:\DOS\DOSKEY.COM
SET TEMP=C:\TEMP
```

When a *DOS World* article instructs you to modify your AUTOEXEC.BAT file, always make a backup copy of the original AUTOEXEC.BAT first. The most common names for your backup copy are AUTOEXEC.BAK or AUTOEXEC.BK. The latter lets you save different versions of your backups—for example, AUTOEXEC.BK1 and AUTOEXEC.BK2. You create a backup copy with the following command:

```
COPY AUTOEXEC.BAT AUTOEXEC.BAK
```

Also, you should have an emergency boot disk available whenever you modify AUTOEXEC.BAT. (See the accompanying section on the facing page, top.) It will let you access your hard drive in case you make an error that locks up your computer. Changes you make to AUTOEXEC.BAT won't take effect until you restart your computer.

BATCH FILES

A batch file is a text file that tells MS-DOS to do a series of tasks. The filename of a batch file always ends with the extension .BAT.

A batch file must be in plain-text format. For example, a batch file might consist of the following lines:

```
CD\
DIR /S /P
```

This batch file moves you to the root directory (CD\) and then gives you a list of all files in all directories (/S), pausing after each full screen (/P).

Every batch file needs a name. In such cases, you should pick your own name. Batch-file names carry the same limitations as any other DOS filename; you're limited to eight characters, plus a three-character extension. A batch-file name must always use the .BAT extension.

To avoid confusion and unexpected results, don't give any batch file the same name as another program or DOS command. For example, VCOPY.BAT is an acceptable name for a batch file, but not COPY.BAT or XCOPY.BAT, because COPY and XCOPY are the names of DOS commands. To run or execute a batch file, type its name at the DOS prompt. For example, to run a batch file called VCOPY.BAT, type VCOPY at the DOS prompt.

Creating and Saving

Using EDIT. If you have DOS 5 or later, you can create a batch file using EDIT. EDIT usually resides in your DOS directory. Type EDIT and enter your batch file. When you're done, press Alt+F and choose the Save option. Type the name of your batch file (make sure you add the extension .BAT) and press the Enter key.

Using other word processors. Most word processors don't save files in plain text; they include other characters, such as control characters that handle such matters as page formatting and typefaces. Most word processors, however, do give you an option to save in plain text. The procedure varies from one word processor to the next. For example, when you save a file in Word-Perfect 5.1, you choose ASCII Text (DOS) as your Format option.

This Magazine

MAKING AN EMERGENCY BOOT DISK

Sometimes a *DOS World* article will suggest that you create a *bootable floppy*— a floppy disk that serves as an emergency system disk. That is, if your computer for some reason can't access your hard drive, you can start your computer from the emergency floppy. You should always have an emergency system disk available, but it's particularly important when you modify AUTOEXEC.BAT or CONFIG.SYS because you may change those files in such a way that your computer won't start from the hard drive. To create a system disk:

1. Insert a floppy disk in drive A.
2. At the command line, type FORMAT A: /S (all existing information on the floppy will be lost).

DOS first formats the floppy disk. Then it copies three DOS system files to the floppy disk: IO.SYS, MSDOS.SYS, and COMMAND.COM.

The first two are hidden files; you won't see them if you type DIR A:. If you have the disk-compression program Double-Space on your computer, the FORMAT command above will also copy DBL-SPACE.BIN, a third hidden file, to the floppy disk.

After you've created your system disk, you should copy a few other basic files to your floppy. Go to your \DOS directory and copy the following files: FORMAT-.COM, EDIT.COM, EDIT.HLP, QBASIC-.EXE, UNDELETE.EXE, CHKDSK.EXE, FDISK.EXE, and SETUP.EXE.

DEBUG SCRIPTS

A Debug script is a list of assembly-language instructions you convert to an executable program using the program DEBUG.EXE in your \DOS directory.

Creating the script. A Debug script must be in plain text. The procedure for creating the script is the same as for creating a batch file. You can use DOS's EDIT program, or you can use a different text editor or word processor and save the script in plain text format.

Creating an executable program. After creating and saving the script, type the following command at the DOS prompt:

```
DEBUG < filename
```

where *filename* is the name of the Debug script you created. For example, if the name of your Debug script is KEYPRESS.SCR, you'd type this line:

```
DEBUG < KEYPRESS.SCR
```

at the DOS prompt. The executable program created by Debug will have the extension .COM. The name of the executable file is determined by the contents of the script. Our convention is to use the same name for the executable file as we do for the script. Thus, the executable file created by KEYPRESS-.SCR will be named KEYPRESS.COM. Once you've created the executable file, you run it by typing its name at the DOS prompt. To run KEYPRESS.COM, type KEYPRESS.

PATHS AND THE PATH STATEMENT

DOS World articles often tell you to make sure that a particular file is in a directory included in your PATH statement. This lets you runs a .COM, .EXE, or .BAT file from any directory on any drive.

For example, an author might tell you to create a batch file called TEST.BAT, put it into a subdirectory called \BAT, and put the subdirectory into your PATH statement. You can then execute TEST.BAT by typing TEST from anywhere on your drives, without having to change to the \BAT directory first.

The PATH statement is a line in your AUTOEXEC.BAT file. It gives DOS a list of directories to search for requested files. Here's an example:

```
PATH=C:\DOS;C:\WINDOWS;C:\BAT
```

When you type TEST at the DOS prompt, DOS looks for the program first in the current directory, then in the root directory, and then, in order, the \DOS, \WINDOWS, and \BAT directories. When it finds TEST.BAT in the \BAT directory, it executes the batch file.

BASIC DEFINITIONS

DOS prompt. Also known as the *command prompt*. By default, the DOS prompt looks like this: C:\>. This is where you type the instructions to run programs or DOS commands.

Boot, boot up, reboot. The process of starting or restarting your computer. Turning on your computer is *booting* or *booting up*. Pressing the key combination Ctrl+Alt +Del restarts, or *reboots* your computer. So does pressing the reset button, if your computer has one.

Extensions. When we refer to a program by its common name (for example, the DOS command FORMAT) without an extension, you can assume that the extension is .COM or .EXE. When we refer to a batch file, we always include the extension .BAT. QBasic program names must always include the .BAS extension.

ASCII. American Standard Code for Information Interchange. For our purposes, an ASCII file is a plain text file, one that consists entirely of

the characters you see on your keyboard.

Directories. Your hard drive has a main directory called the *root* or *home* directory. Directories created off the root directory are called *subdirectories*. When we provide the name of a subdirectory, it will look something like this: \WORD\FILES. Here, the root directory has a subdirectory called WORD, which in turn has a subdirectory called FILES.

File placement. We assume that the following files are in your root directory: AUTO-

EXEC.BAT, CONFIG.SYS, and COMMAND.COM. We also assume that your DOS files are in a DOS subdirectory, usually called \DOS.

Keystroke combinations. When you should hold down one key while pressing a second, we indicate it this way: Alt+F4 (press the Alt key and hold it down while you press the F4 key). When you should press one key, release it, and press another, we indicate it this way: Alt, F4 (press the Alt key, release it, then press the F4 key).

QBASIC PROGRAMS

QBasic is the programming language included in all versions of MS-DOS since version 5. The name of a QBasic program always ends with the extension .BAS.

Typing in the listing. Type QBASIC at the DOS prompt and press Enter to start. Now type in the listing as printed, pressing Enter at the end of each line. Note that when a line in the listing is indented two spaces from the line above and doesn't start with a command or keyword, it's a continuation of the previous line. Other indentations, or none at all, indicate a new line.

Subroutines and functions. QBasic listings often include subroutines and functions, and typing them is confusing at first. They begin with a line containing the keyword SUB or FUNCTION. Note that when you type a SUB or FUNCTION line and press Enter, all other lines you've typed will disappear from view. This can be disconcerting for beginning programmers. There's nothing to worry about—your listing is safe. To avoid screen clutter, QBasic simply hides other parts of your listing when you're typing in a subroutine or function. To see the other parts of your program, open the View menu at the top of the QBasic screen, then select SUBS. The SUBS dialog box will appear, letting you select the part of the program listing you want to view.

Saving a listing. Save your partially completed listing as you go along, rather than waiting until you've typed in the whole thing. To save, open the File menu, choose Save, and type in a filename when QBasic prompts you. We suggest using the filename specified in the magazine article. Subsequent saves of your listing won't prompt you for a filename, but will instead use the filename indicated the last time you saved the listing.

Running a program. After you've typed in the entire listing and saved it a final time, you can run the program by selecting Start from the Run menu or pressing Shift+F5. If QBasic finds an error, it will stop the program and highlight that line. To run a QBasic program (a .BAS file) stored on your hard drive, start QBasic, then select Open from the File menu. Choose from among the .BAS files displayed in the open dialog box to load the program into QBasic, then select Start from the Run menu or press Shift+F5. To stop a QBasic program, press Ctrl+Break; select Exit from the File menu to return to DOS.

DOS World BBS. Typing and debugging a long listing is time-consuming. If you have a modem, our listings are always available on *DOS World*'s bulletin-board system (BBS) at 603-924-3181. There are no connect-time charges; you pay only for the phone call. Set your communication program to 8 data bits, no parity, 1 stop bit (8, N, 1). Dial the number and wait for the "Connect" message. If you're a first-time user, the system will ask you to enter your name and choose a password. Then it will display a general information screen, followed by a questionnaire requesting your address, phone number, and so on, so that we may set up your account. From this point, on-screen prompts are the same for all users. A series of messages present the latest BBS news; press Enter after each message to go to the next screen. From the Bulletin Menu, Bulletin #1 offers information on navigating the Main and File Menus, with instructions for listing, marking, searching for, and downloading files. ■

DOS WORLD

PowerBASIC 2.1

Favorite Shareware Utilities

MISCELLANEOUS

CAREER OPPORTUNITIES

Computer Analyst/Programmer $32,000.00 per year. Immediate opening within a firm located in Memphis, TN. Will be involved in application program software design and development, using Unix/C and MVS/CICS systems. More specifically, will develop in-house data communcations software and design databases to support it, utilizing user interface, file and data structures. Will also utilize TCP/IP and other system resources to implement cross-platform data transmitting and processing between UNIX and MVS systems on IBM Mainframe. Must possess a MS in Computer Sciences which included at least one academic year of coursework or practical experience in C programming language and the Unix Development Environment, and one year of professional experience in software design and development which included use of IBM mainframe, MVS, SQL and CICS. Graduate level program must also have included one course each in Algorithm Design and Analysis, Database Design, Operating Systems, Data Communications and Network Design. Send resume to Kathy Malo, Job Service Program & Technical Support, TENNESSEE DEPT. OF EMPLOYMENT SECURITY, Nashville, TN 37245-1200, Ref. Job Order No. TN1477121. EOE

Systems Analyst/Programmer $30,000.00 per year. Immediate opening within a firm located in Memphis, TN. Will perform in-house systems analyses; maintenance and up-gradation of existing store management systems; Wide Area Networking (WAN) applications; Database systems on a multiplatform networked operating environment using "C" programming language and state-of-the-art equipment and tools. Coordinate releases to company stores throughout the U.S. from the firm's Memphis headquarters and develop new services to support individual outlet and customer needs. Must possess a B.S. in Computer Science with 6 month full or part-time experience in programming. Experience and education must have included programming in the Unix operating system, networking, analysis, diagnosis and solution of software problems. Send resume to Kathy Malo, Job Service Program & Technical Support, TENNESSEE DEPT. OF EMPLOYMENT SECURITY, Nashville, TN 37245-1200, Ref. Job Order No. TN1477147. EOE

POWERBASIC TOOLS

Say you saw it in the
DW Marketplace

MISCELLANEOUS

VIDEOS

DOS Watch

News and reviews of DOS enhancements, upgrades, and products

edited by Steven F. Smith

How to Improve Your Image

NeoPaint version 3, recently released by NeoSoft Corporation, is one of the few updated graphics applications designed to run strictly under DOS, or as a DOS application under Windows or OS/2.

Enhancements to this paint and image-editing product include an oil-paint tool, a quill pen, and a free-form 3D tool. Watercolor and oil-brush tools allow retouching of existing images with natural-looking strokes.

Sold through retail outlets as **NeoPaint Presentations**, the package includes NeoPaint, a screen-capture utility, and a presentation application for creating slide shows. System require-ments are 286 PC or better, DOS 3.1 or later, EGA/VGA/Super VGA or compatible graphics card and monitor, and Microsoft-compatible mouse. A Sound Blaster or compatible sound card is optional.

NeoPaint, $69.95; upgrade for registered users, $20; NeoSoft Corp., 354 N.E. Greenwood Ave., Suite 108, Bend, OR 97701-4631; 800-545-1392; fax 503-388-9221; BBS 503-383-7195.

The Complete Money & Home Manager, from KEBA International, is billed as ten DOS programs in one to help **Penny-Pincher** you keep your home and money organized. This package will track your credit information, vehicle maintenance, and home-improvement progress; keep you informed of your net worth, assets, liabilities, and budget; and catalogue your CDs, recipes, books, collec-tions, and other personal property.

CM&HM is designed to run on any IBM compatible PC with at least 3MB of available storage space and DOS 3.1 or later.

The Complete Money & Home Manager, $29.95; KEBA International, P.O. Box 15131, Columbus, OH 43215-0131; 800-847-4561.

Tapedisk Corp. has developed a software package for SCSI tape drives (such as QIC and 4mm DAT) that lets your system write data directly from any DOS or Win-dows application to the storage device—without having to restore the file to the disk before use.

Tapedisk 6.2, billed as the only current DOS/Windows product to treat a

Direct Deposit

SCSI-format tape drive like a disk drive, is a 40K to 45K DOS device driver (which you can load high), installed in your CONFIG.SYS file. Each package includes free per-sonal storage-management software valued at $79.

Tapedisk 6.2, $249.95; Tapedisk Corp., 816 Meadow Hill Drive, Menomonie, WI 54751; 715-235-3388; fax 715-235-3818.

FROM THE BOOKSHELF: TECH HELP AT YOUR FINGERTIPS

For any DOS user who's been stumped by nagging soft-ware problems, Osborne/McGraw-Hill and Corporate Software Inc. have teamed up to produce **DOS Answers: Certified Tech Support**, one of a series of solution books designed to save computer users time, money, and stress. Corporate Software is the world's largest third-party tech-nical support provider, and Osborne has culled the firm's data bank to produce answers to the most commonly asked questions.

Included are the top ten tech-terror tips to help you solve the most common problems encountered by DOS users; words of widsom for all users who want to prevent problems before they rear their ugly head; and extra tips offering optional methods of attacking a problem. The Osborne/Corporate Software series also targets Lotus Notes, WordPerfect for Windows, Word for Windows, Excel for Windows, and NetWare.

DOS Answers: Certified Tech Support, $16.95; Osborne/McGraw-Hill Inc., Blue Ridge Summit, PA 17294-0850; (800) 722-4726.